SCORE 7
SHIPS 2
HI-SCORE
LEVEL
6
10
0 COM
TIME 770
HI-SCORE 0
SCORE 0
ワナノカズ 3
トクテン 0
MARIO'S 100M RUNNING
PERFECT
SCORE: 449
ファミリーベーシック™
大百科（だいひゃっか）

これが ファミリーベーシック!!

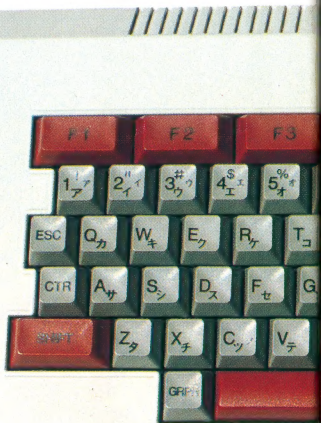
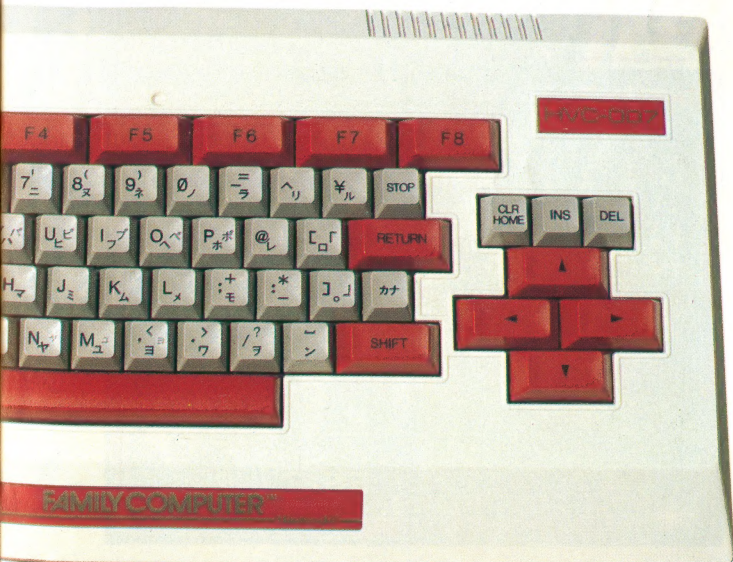
誰でも簡単に扱えるパソコン——それがファミリーベーシックだ。単なるゲーム遊びにあきちゃったら、こんどは自分でゲームを作ってみよう。サウンド機能やグラフィック機能も充実しているから、本格的なゲーム作りが楽しめるぞ！

ベーシック V2

いま人気爆発中のカセットだ。ミュージックボードなどの機能がついているぞ！

▲これがV2のカセットだ

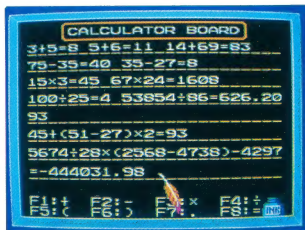
▼ファミリーベーシックにとって必要不可欠なキーボード。これでプログラムを入力するんだ

ベーシック(ブイスリー)V3

▲V(ブイスリー)3のカセットだよ！

いろいろなベーシックの

ファミリーベーシックにはゲームベーシック面のほかに、3つの画面が用意されているぞ。自分の好きな画面を選んで、手がるに音楽や占いを楽しんじゃおう！

カリキュレーターボード

CALCULATOR BOARD
3+5=8 5+6=11 14+69=83
75-35=40 35-27=8
15×3=45 67×24=1608
100÷25=4 53854÷86=626.20
93
45+(51-27)×2=93
5674÷28×(2568-4738)-4297
=-444031.98
F1:+ F2:- F3:× F4:÷
F5:(F6:) F7:. F8:=

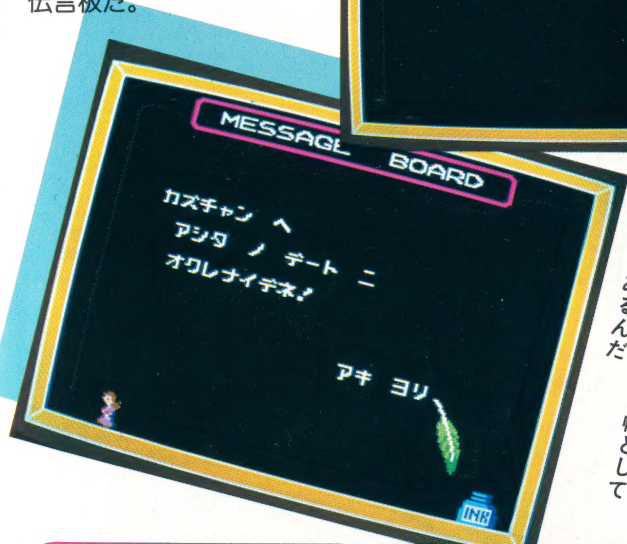
計算式を入力すればコンピュータが答えてくれる計算ボードだ。

ミュージックボード

キミが入力した音楽をコンピュータが自動演奏してくれるぞ。

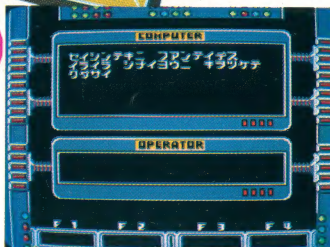
◀キミの好きな曲を入力しよう。なん回でもくり返し演奏してくれるぞ

遊び方

メッセージボード

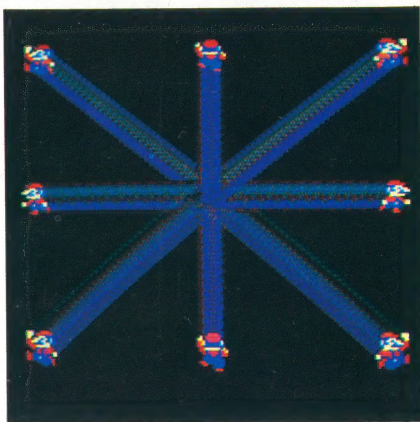
口に出していいにくいことをコンピュータが代弁。メッセージボードは家庭の伝言板だ。

▲伝言板以外にも、記録帳としての使い方もあるんだ

コンピュータ占い

キミの生年月日を入力すると、コンピュータがキミの運勢をバッチリ占ってくれるぞ。

ゲームベーシックはおもしろい

マリオの8方向移動

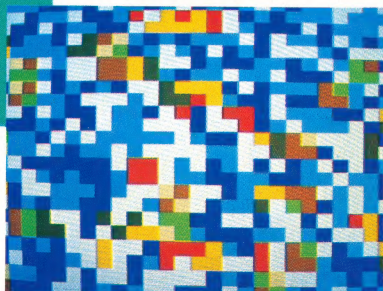
MOVE命令を使って8人のマリオをそれぞれ別の方向に移動させてみよう。

マリオばかりじゃなく他のキャラクタもいっぱいいるぞ。

```
LIST
0 CLS
10 FOR I=0 TO 30
20 LOCATE RND(27),RND(23)
30 PRINT CHR$(205+RND(2))
40 NEXT
50 GOTO 50
OK.
```

ランダムグラフィック

BGグラフィック画面を使えば、背景を描いたり、キャラクタに色を着けたり、楽しいことがいっぱいできるぞ。

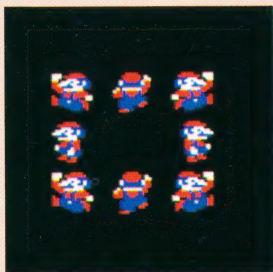
大好きな歌をベーシックが演奏

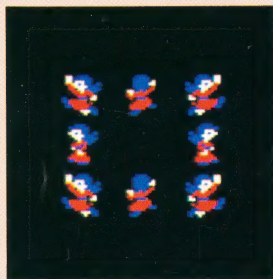
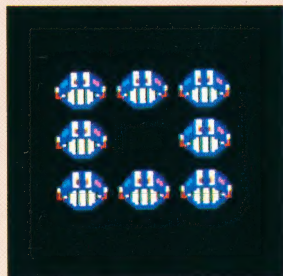
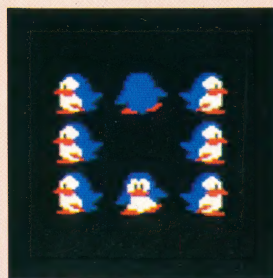
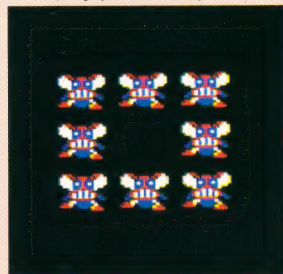
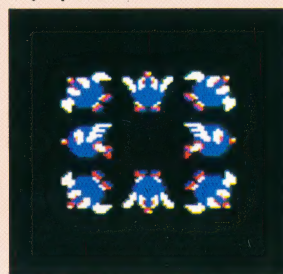
```
200 PLAY"EEEE:R01A:O0A3ACA
210 PLAY"EDCDEAAA:RARE:AAEAA
220 PLAY"AGGGFF:RER:AACA01DD
230 PLAY"FFFEDF:ARA:DDDDDD
240 PLAY"F8R3D:RARA:DDDDDDDD
250 PLAY"DDDDDC:RGR:O0GGBGGG
260 PLAY"O1BO2CDGGG:GRG:BGGGBG
270 PLAY"G7C3CC5:RGE3E6:GGBGO1C
280 PLAY"D3DD5E:#F3#F6R5:D3D6E3
290 PLAY"F3E7R3:#GR#G:BEEEBE
300 PLAY"R7:O3E3DFE:O1EDFE
310 PLAY"R6E3EE:RG5A5R3:RG5A3O0AA
```

ミュージックボード画面じゃものたりないっていう人は、ベーシックで音楽を入力してみよう。和音やエンベロープも楽しむことができるぞ。自分の好きな音色まで選ぶことができるからゴキゲンだ。

```
490 PLAY"R7:C602C3:R7
500 PLAY"F3FFF:C601C3:R7
```

全(ぜん)キャラクタの

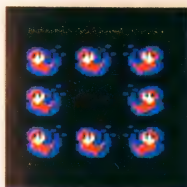
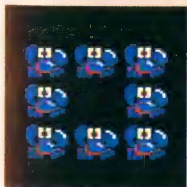
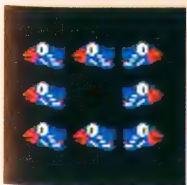
▲マリオ

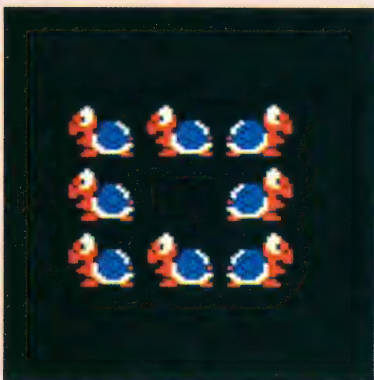
▲レディ

▲ペンペン

▼ニタニタ

▼ファイターフライ

▼アキレス

カラー表

ファミリーベーシックにはマリオ、レディなどの人物から、スターシップ、スターキラーなどの兵器にいたるまで、全部で16種類のアニメキャラクタが用意されているぞ。それぞれのキャラクタにどんなポーズをとらせるかはキミのおもいのまま。ゲームをプログラムする上で、必要に応じて使い分けよう。

▲ファイアーボール

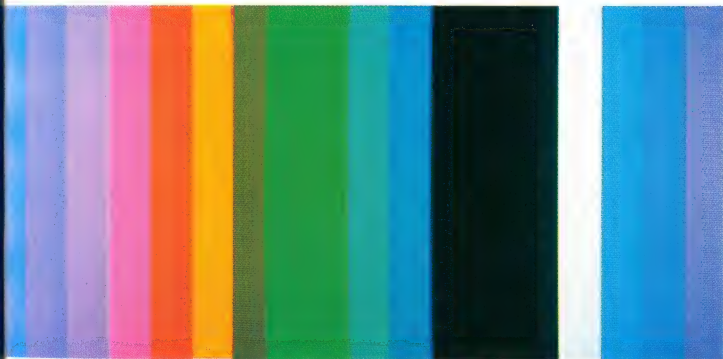
▼車

▲スターシップ

▲サイドステッパー

▲ニットピッカー

▲シェルクリーパー

52色カラーチャート

これがファミコンのカラーチャートだ。ファミリーコンピュータでは全部で52色の色を出すことが出来るんだ。チャートの数字が、52色以上になっているのは、色のだぶりがあるためで、じっさい見てみると、52色見られ

るはずだ。

パレットで色指定をする時は、表の下のコード番号を合わせてやることによって所定の色を出せるんだ。

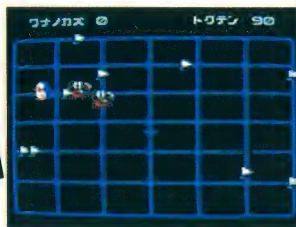
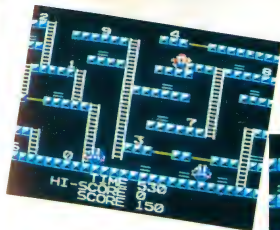
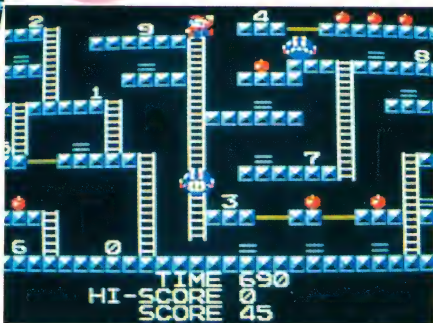
文字のキャラクタやスプライトのキャラクタが少ないなんて言っている人は、スプライト命令と、カラーチャートを使ってオモシロキャラクタを作ってみよう。

V3 内蔵プログラムで遊ぼう!!

ファミリーベーシックV３には、最初からすでに４つのゲームが内蔵プログラムされているんだ。キミの好きなゲームを選んで遊ぼう。改造も簡単にできるよ。

ハート

コントローラーのマイクに声や息を吹きかけて、画面にハートのマークを描こう。ハートが完成し、画面中央でマリオとレディが出会えばOK！

ペンペン迷路

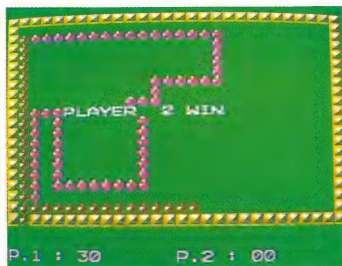
ペンペンを上下左右に移動させて、迷路にある数字を取っていこう。カニさんにつかまるとアウトだぞ。

マリオワールド

マリオに数字やリンゴを取らせるゲームだ。でもニタニタがしつこく追いかけてくるから、気をつけなきゃいけないぞ。

ハシゴやジャンプ台の使い方など、かなりのテクニックが必要だ。

スターキラー

宇宙船からミサイルを発射して、敵の飛行体をやっつけよう。友だちと2人でプレイすることもできるぞ。

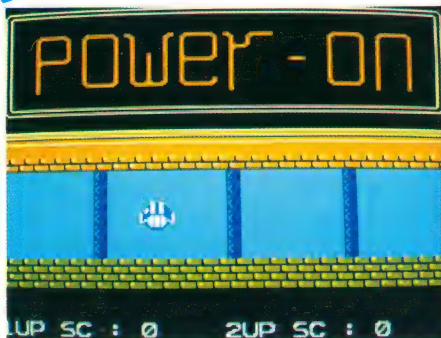
オリジナルゲームを

1 アステロイドクラッシュ

襲い来る隕石をよけながらスターシップを進めていくゲームだ。

2 りんご陣取りゲーム

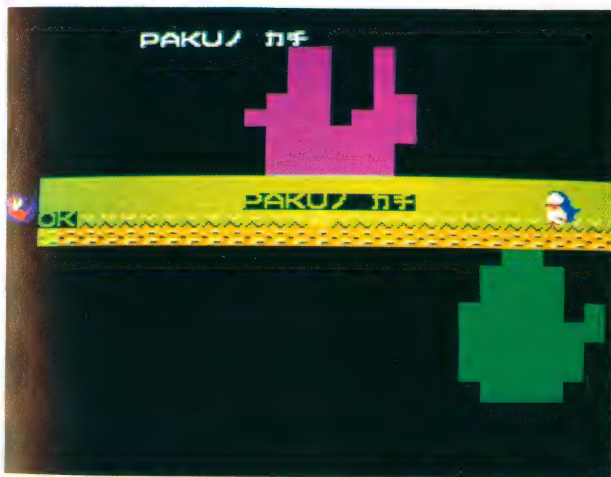
友だちと2人で遊ぶゲームだ。画面にリンゴをいくつも並べていこう。先に四方のカベに突き当たった方が負けだ。かけひきを必要とするゲームだよ。

3 ニタニタプッシュ

これも2人用のゲームだ。画面中央に姿を現わしたニタニタを、相手の陣地に先に押しこめた方が勝ちだ。

作ろう!!

キミだけのオリジナルゲームを作っちゃおう!

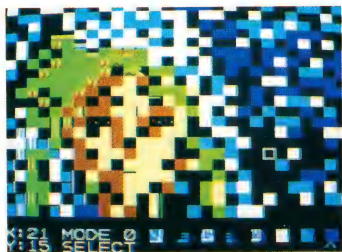
4 ジャンケンポン

コンピュータを使ったジャンケンゲームだ。

◀ジャンケンに勝つと自分のアニメキャラクタがどんどん進んでいくんだよ

5 マリオの100mランニング

コントローラーのA・Bボタンを交互に押してマリオを走らせよう。キミのマリオは100mを何秒で走れるかな。ベストタイムが出るとファンファーレ。

BGグラフィック

▶女の子のイメージで…

BGグラフィック画面を使うと、画面にキミの好きな絵を自由に描くことができるぞ。着色も自由自在。いろんな絵を描いてみよう。

◀ビデオアニメ〝メガゾーン23〟の超戦闘メカ「ガーランド」を描いてみたぞ!!

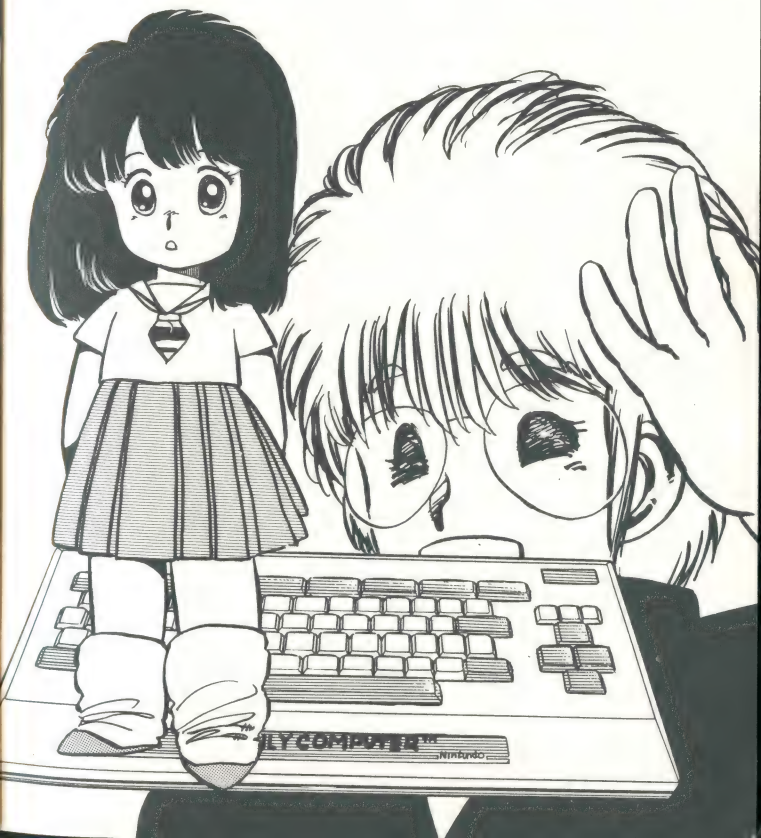
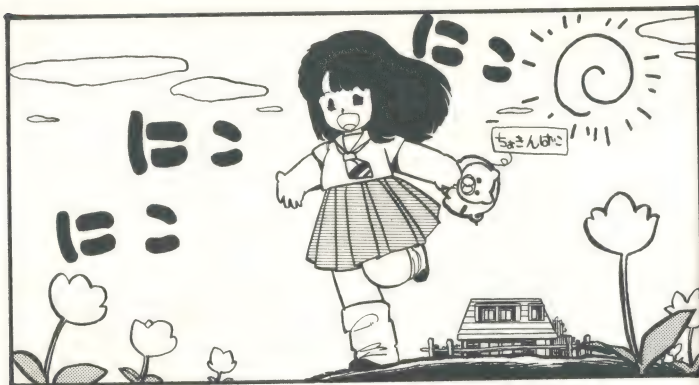
ファミリー
ベーシック

Contents.

もくじ

カラー口絵（くちえ）……1
これがファミリーベーシック!!……2
いろいろなベーシックの遊び方（あそびかた）……4
ゲームベーシックはおもしろいぞ!!……6
全（ぜん）キャラクターのカラー表（ひょう）……8
52色（しょく）カラーチャート……10
Ｖ３（ブイスリー）・内蔵（ないぞう）プログラムで遊（あそ）ぼう!!……12
オリジナルゲームをつくろう!!……14
ＢＧ（ビージー）グラフィック……16
マンガ・ベーシックで遊（あそ）ぼう!!……19
キーボードの操作（そうさ）……31
カリキュレーターボード……32
ミュージックボード……34
メッセージボード……38
コンピュータうらない……40
ゲームベーシック……41
マンガ……42
スプライト・オン……47
一般的（いっぱんてき）な命令（めいれい）……57
プログラムのための命令（めいれい）……65
画面制御命令（がめんせいぎょめいれい）……73
音楽用命令（おんがくようめいれい）……81

関数（かんすう）……85
特殊な命令（とくしゅなめいれい）……91
メモリー制御命令（せいぎょめいれい）……97
SAVE・LOAD（セーブ・ロード）……101
マンガ……107
ベーシックのベーシック……119
基本命令（きほんめいれい）……125
MOVE命令（ムーブめいれい）……139
BGグラフィック（ビージー）……155
ミュージック……181
まとめ……197
エラーメッセージ……213
V3・内蔵プログラム（ブイスリー・ないぞう）……221
オリジナルゲームをつくろう!!……237
コマンド表（ひょう）……269

構成・レイアウト　ページワン
文　京極狂介
マンガ　ラジカル・カンパニー／眞樹亜美霞
イラスト　眞樹亜美霞／竹川雄二／かぐや姫
PHOTO　スタジオキャッツ　ハート・ビート
サンプル・プログラム　かりあげやっちゃん
ミュージック・プログラム　♡ぱ　く
監修　吉岡真太郎

ベーシックで遊(あそ)ぼう!!

眞樹亜美霞(まきあみか)

にこ
にこ
にこ
ちょきんばこ

今、話題の
ファミリーベーシック

わーっ
すっごい人気

メイ・加藤

ありがと
はい
おじょうさん
ぼくも
ほしーな
うらやましー
すっごー
わーあの子
ベーシック
買った!!
ウフフ
かずよちゃん家

さっそく
開始!!

オン
ピ
ワタシハ
ファミリーコンピュータ
デス
ピ
ピ

わぁっ!!
COMPUTER
アナタハ ダレデスカ?
ナマエヲ イレテ クダサイ

かずよ♡

ピッ

ピッ

ピ

ピ

ピッ

ピ

リョウカイ!!

アナタハ
カズヨ　サン　デスネ

どれにしよーか
まよっちゃうな

決めた!!
カリキュレーター
ボードにしよっと!!

え

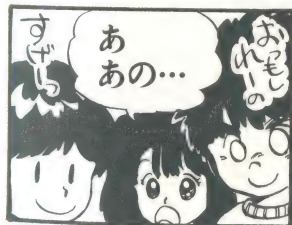
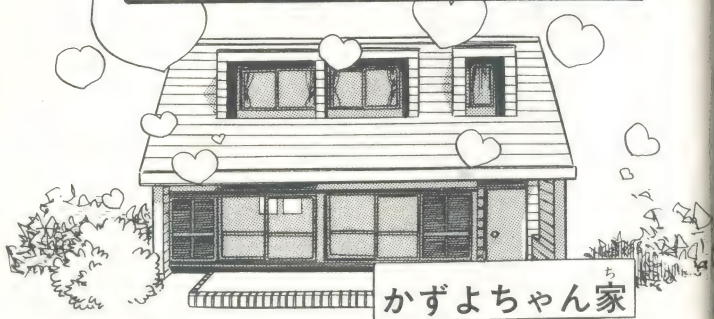
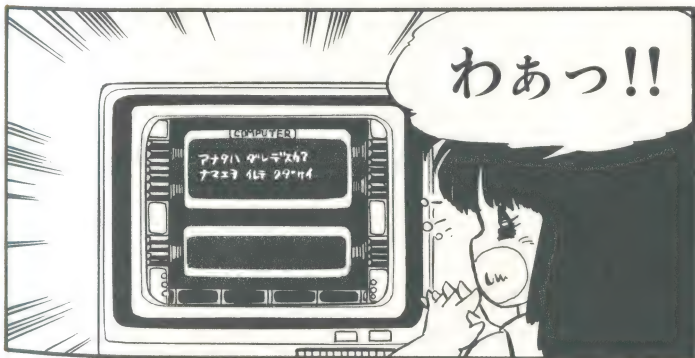
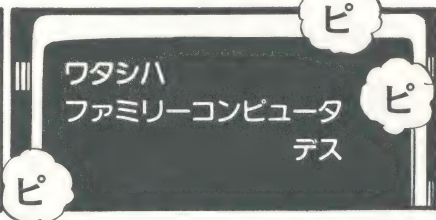
カリキュレータ
ーボードって
計算なのかあ
わたし
計算って
ニガテ
ファミリー
ベーシック
マニュアル

う〜んと
カンタンなのに
しーよう♡

1+1

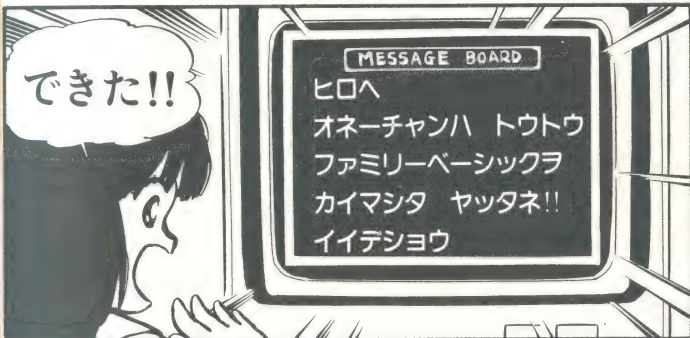
2

すっごぉーい!!
ベーシックが計算
してくれちゃうんだ

(1 ×100−10)÷3

さほど　むずかしくない?

(1 ×100−10)÷3= 30

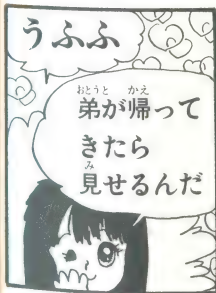
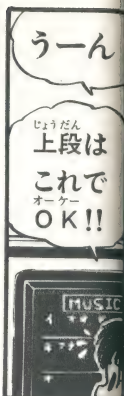
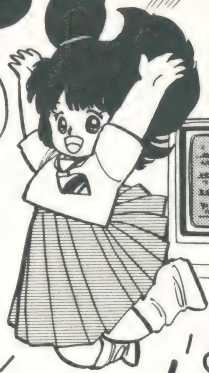
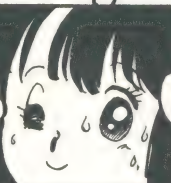
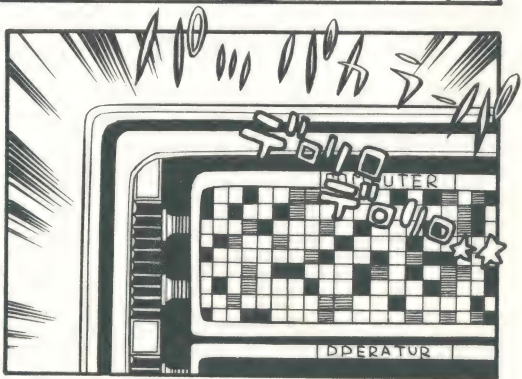
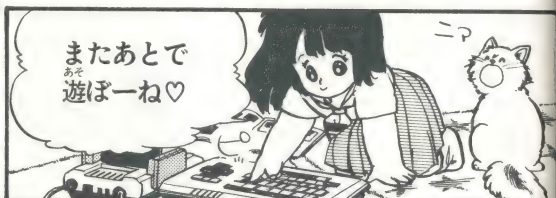
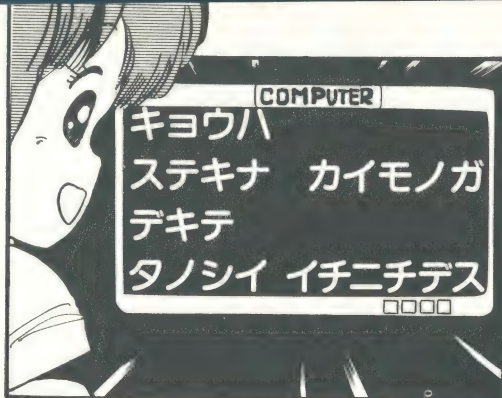
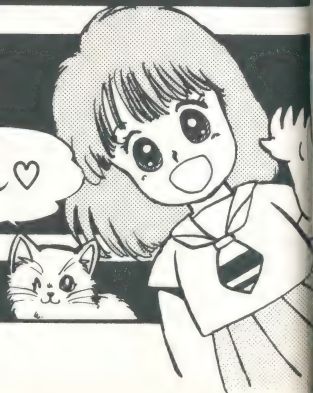
はやーい!!
ばっちりー!!

これなら
あしたの算数の
宿題
バッチリよ!!

次は
メッセージ
ボード!!

えーと
伝えたい言葉を
キーで入力すれば
いいのね
BOKU PERU DESU
YOROSHIKU ♡

できた!!
MESSAGE BOARD
ヒロヘ
オネーチャンハ　トウトウ
ファミリーベーシックヲ
カイマシタ　ヤッタネ!!
イイデショウ

うふふ
弟が帰って
きたら
見せるんだ
うらやましがるぞ
きっと!!
うふふ、
すごーい
おねーちゃん
あーっ!!
エヘン!

うーん
上段はこれでOK!!
MUSIC
ミ
ド
ド
ド
次はミュージック
ピ
ピ
スポ
INK
なかなかムズカシイけれど
中段 下段できあがり

自動演奏
すると
最高!!
よーし
今日の日を
うらなって
もらお♡
ベーシック
買って
よかった!!
わたしは
3月3日
生まれで……
今日は
10日
パッパカラ~
デロリロ
デロリロ
COMPUTER
OPERATUR

サギョウヲ　チュウシ　シマス

各種キーの使い方をおぼえよう

キーボードの操作

ファミリーベーシックに必要不可欠な存在がキーボードだ。なぜって、プログラムの入力はすべてこのキーボードを使っておこなうんだからね。各種キーの使い方をマスターしよう。

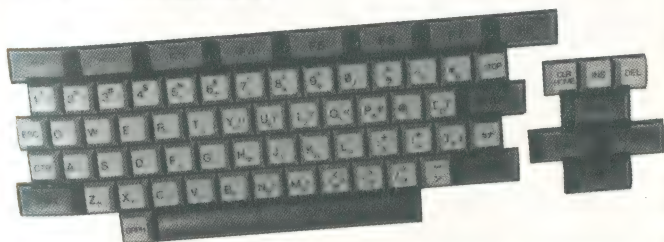
■文字キー
数字、英文字、カタカナ、記号を入力するためのキーだ。全部で48個ある。

■SHIFTキー
文字キーとの組み合わせで、カタカナの小文字やキーの上と右に書かれた記号を入力。

■RETURNキー
これを押すと、入力した文字や記号がメモリされ、カーソルが次の行の先頭に戻る。

■　スペースキー
文字と文字の間に1文字分のスペースを作るキー。

■◀▶▲▼カーソルキー
文字や数字の入力する位置(カーソル)を上下左右に動かすキーだ。

■GRPHキー
文字キーと組み合わせてカタカナの濁音を入力。またグラフィック記号を入力。

どんな計算でもバッチリこなす

カリキュレーターボード

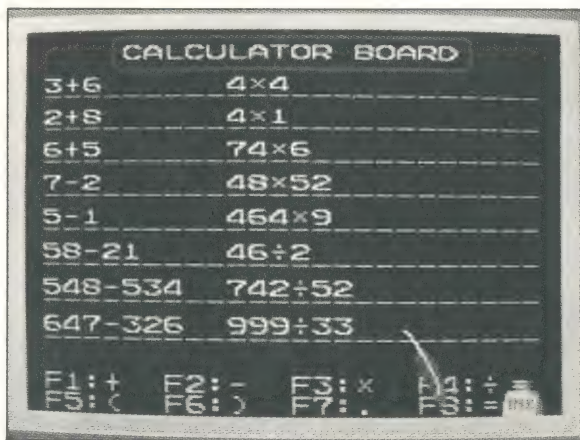
カリキュレーターボードは数字計算のための画面だ。いわば電卓の役目をもっているといえるね。

使い方は、まずキーボードで計算式を入力。するとそれにしたがって羽根ペンが画面に式を書いてくれるんだ。そして、式の後に＝(イコール)を入力すると答えが出るぞ。

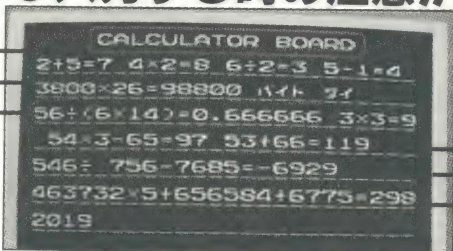
カリキュレーターボードでは、画面に入る字数はヨコ24字×タテ8行。1行でいくつもの計算ができるし、また2行以上にわたる計算も可能だ。8ケタまでの＋、－、×、÷による四則計算ならなんでもござれだ。

CALCULATOR BOARD

3+6	4×4
2+8	4×1
6+5	74×6
7-2	48×52
5-1	464×9
58-21	46÷2
548-534	742÷52
647-326	999÷33

F1:+　F2:-　F3:×　F4:÷
F5:(　F6:)　F7:.　F8:=

◀カリキュレーターボードで、算数ドリルを作っちゃおう！

●入力する時の注意!!

```
CALCULATOR BOARD
2+5=7 4×2=8 6÷2=3 5-1=4
3800×26=98800 バイト サイ
56÷(6×14)=0.666666 3×3=9
 54×3-65=97 53+66=119
546÷ 756-7685=-6929
463732×5+656584+6775=298
2019
```

符号につづく－の数字はカッコでくくるんだ。

計算式といっしょにメッセージも入れられるぞ。

1行の中にいくつでも計算式を入れられるんだ。

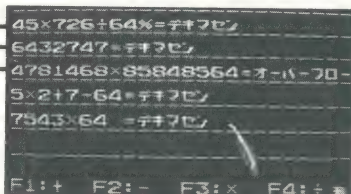
2行以上にわたる長い計算だってできるんだ。

ひとつの計算式の中には、すきまを作らないこと。

ひとつの計算式がおわったら、1文字分あけて次へ。

●これはエラーになるよ!!

```
45×726÷64%=デキマセン
6432747=デキマセン
4781468×85848564=オーバーフロー
5×2+7-64=デキマセン
7543×64 =デキマセン

F1:+  F2:-  F3:×  F4:÷
```

計算式の中に指定の符号や数字以外の文字や記号を入力したとき。

計算式になっていないものを＝で結んだときはエラーだ。

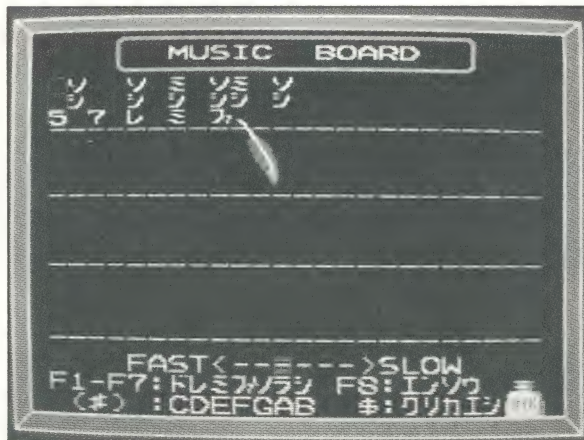
答えが8ケタを越えたときは、オーバーフローと表示される。

前行の計算との間に、スペースがとられていないとき。

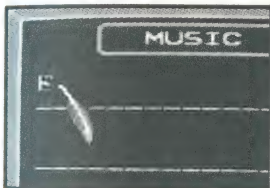
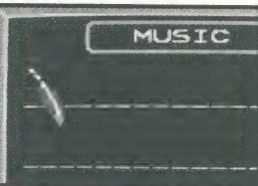
コンピュータで音楽演奏！
ミュージックボード

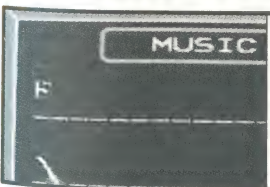
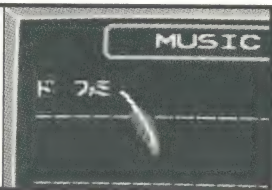
ミュージックボードは、キミのコンピュータに音楽機能をもたせるための画面なんだ。だから、これを使えば、キミのコンピュータが楽器に早替わりしてしまうっていうワケ。

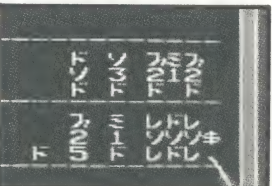
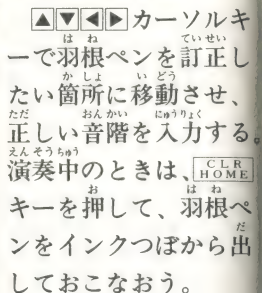
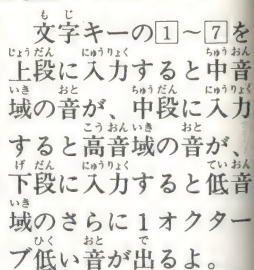
まず、キミの好きな曲の音階をドレミ～で入力する。入力欄は、３音×24マス×４行あるから、そこに収まる長さの曲ならＯＫだ。音階は上段、中段、下段にわかれていて、３オクターブの音を同時に出すことだってできるぞ。

入力がおわったら、こんどは演奏させてみよう。なん回でも繰り返し、自動演奏してくれるぞ。

作曲も自由自在。マイクをつなげば、カラオケマシンだ。

●音階の入力方法

最初、羽根ペンは１行目の上段、１音目にある。どのオクターブでいくか◀▶▼▲のカーソルキーで羽根ペンを入力したい位置に移動。

位置が決まったら、ファンクションキーで音を入力する。［Ｆ１］～［Ｆ７］はド～シに対応するぞ。［ＳＨＩＦＴ］キーを使うと半音上がるよ。

音が入力されると、羽根ペンは右へ１音分移動する。音を入れたくないときには［　　　］スペースキーで空白をつくろう。

１行目の入力がおわると、羽根ペンは次の行の１音目へ移動するぞ。行の途中でも［ＲＥＴＵＲＮ］キーを使えば改行できるよ。

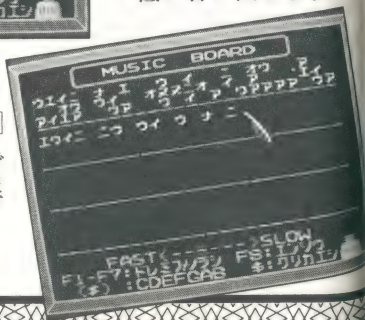
くり返し曲を自動演奏させたい場合は、曲の最後にマーク＄を入力しておこう。上、中、下段、どの位置に入力してもかまわないよ。

●音階を修正したいときは…

▲▼◀▶カーソルキーで羽根ペンを訂正したい箇所に移動させ、正しい音階を入力する。演奏中のときは、CLR HOMEキーを押して、羽根ペンをインクつぼから出しておこなおう。

もっとたくさん音階を使いたいナ…

文字キーの1～7を上段に入力すると中音域の音が、中段に入力すると高音域の音が、下段に入力すると低音域のさらに1オクターブ低い音が出るよ。

半音も出せるよ!!

カタカナのアイウエオナニを入力すると、1～7で入力した音より半音高い音が出るぞ。

●さあ、演奏開始だ!!

◀キミの入力した曲をいよいよコンピュータに演奏させてみよう！

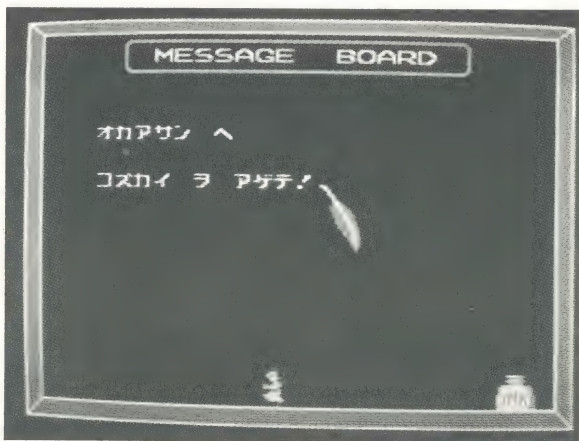
①F 8のファンクションキーを押すと、羽根ペンがインクつぼに入り、演奏が始まるぞ。
②演奏している箇所は、カーソルが右に移動して知らせてくれる。また、上、中、下段の音は同時に演奏されるぞ。
③演奏中に◀▶カーソルキーを押すと、ＦＡＳＴ <−−−−−−> ＳＬＯＷのカーソルが移動、演奏の速さが変化するぞ。◀だと速くなり、▶だと遅くなるんだ。

④　　　　　スペースキーを押すと演奏は中止される。このとき、演奏スケールの位置はそのままだけど、◀▶キーを押すごとに、演奏スケールは１マスずつ左右に移動する。演奏を再開したいときは、F 8キーを押せばＯＫだ。
⑤IIコントローラーについているマイクロホンを利用すれば、カラオケを楽しむこともできるぞ。みんなでカラオケ大会をひらこうぜ。

いいにくいこともこれでOK！

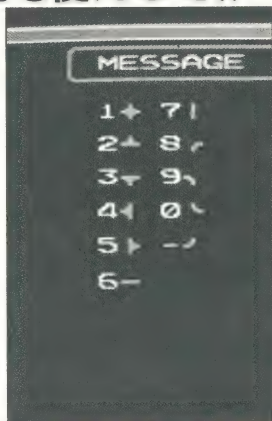
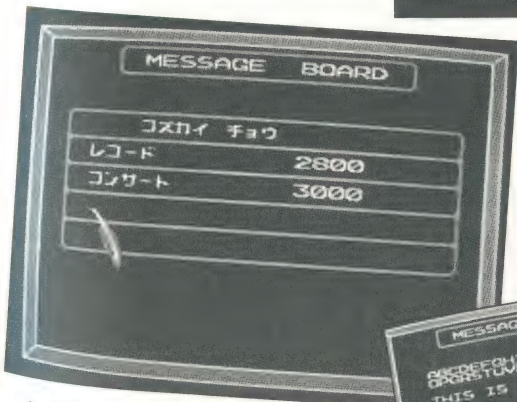
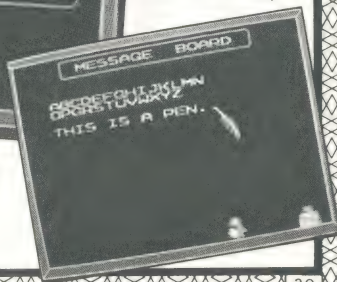
メッセージボード

ケンカした友だちと仲直りしたい、パパやママにおねだりをしたい……、こういうことって口ではなかなかいえないもの。でも、そんなときはメッセージボードを使っちゃおう。ベーシックのカセットについているメモリバックアップスイッチを使ってメッセージを記憶させておけば、コンピュータがキミのいいたいことを代弁してくれるぞ。

書ける文字数は、ヨコ24文字×タテ17行。画面をすべて消したいときは、SHIFT キーを押しながら CLR HOME キーを押せばOKだ。

◀マリオやレディも登場して、キミのいいたいことを伝言してくれるぞ

●グラフィック機能も使えるぞ!!

メッセージボード画面では、カナ状態のとき、GRPHキーを押しながら文字キーの最上列を押すと、13種類のグラフィック記号を描くことができるぞ。

このグラフィック機能を使って、こずかい帳やお買い物リストを作るのもいいよね。メッセージボードのいろいろな使い方を考えてみよう。

▼キーボードの英文字、記号はタイプライターと同じキー配列。文字の入力練習をしよう

▲グラフィック機能を使って作ったこずかい帳。とてもわかりやすくて便利だね

キミの運勢がすぐわかる！

コンピュータうらない

キーボードで、[H][E][L][L][O]、[オ][ハ][ヨ][ウ]、[コ][ン][ニ][チ][ハ]、[コ][ン][バ][ン][ハ]、[F][3]のどれかを入力すると、コンピュータがあいさつをし、占いをしてくれるぞ。

①まず、生年月日を入力しよう。年、月、日の間にはスペースを。

②次に、占ってほしい日を入力する。

③すると、その日が生まれてから何日めかがわかるぞ。

④ファンファーレとともに占いの結果が出るぞ！

GAME BASIC
ゲームベーシック

うーん
うーん
うーん

ゲーム
ベーシックで
遊びたいん
だけどぉ
プログラムの
作りかたが
わかんない

そーだ!!
おかーさんに聞こう!

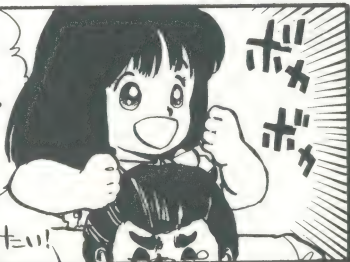
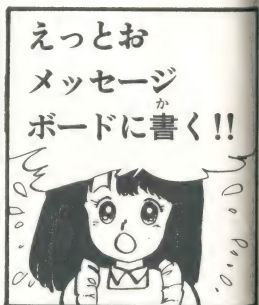
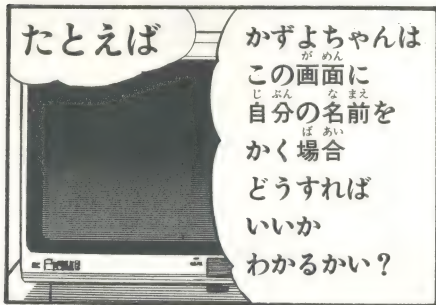
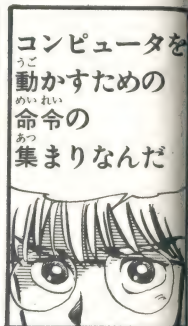
おかーさん
ベーシック
にこっ

おとーさん
ベーシック

おほほっ
たたたっ
千葉さん注目!!

もーっ!!

ぱった
死んだふり

ボカ
ボカ
そーだ!!
近所の吉岡くんに
聞いてみよー
っと!!
いたい!

吉岡くん家

コンピュータを
動かすための
命令の
集まりなんだ
ベーシックと
いうのはねー

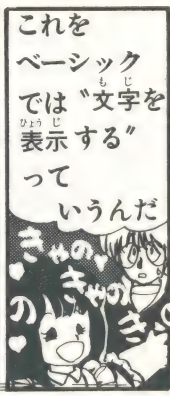
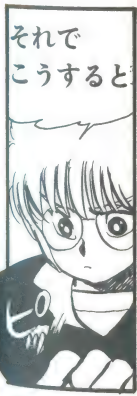
えっとお
メッセージ
ボードに書く!!
かずよちゃんは
この画面に
自分の名前を
かく場合
どうすれば
いいか
わかるかい?
たとえば

ずるけ…
そんなことぐらい
知ってるよーだ!!

ほら

そーじゃなくて
ベーシックでは
PRINTと言う
命令を使って
書くんだよ

へー

これを
ベーシック
では“文字を
表示する”
って
いうんだ

そういった命令と組み合わせることをプログラムというんだよ

わかった?

うん!!

かずよわかんない!!

ゲームベーシック

スプライト・オン

```
LIST
10 DATA "12345"
20 DATA "&HABC"
30 DATA "-123 "
40 DATA "+14  "
50 DATA "POS  "
60 FOR I=1 TO 5
70 READ A$,B$
80 PRINT A$ ; " --> " ; VAL(A$)
90 NEXT
OK
RUN
12345 --> 12345
&HABC --> 2748
-123  --> -123
+14   --> 14
POS   --> 0
OK
```

スプライト制御文を使って アニメキャラクタを動かそう!

この項目で出てくる命令語

SPRITE ON　DEF MOVE

MOVE　POSITION　CUT　ERA

SPRITE　XPOS　YPOS　VCT

CAN　CRASH

スプライトの表示指定

スプライトでキャラクターを動かすためには、まずＳＰＲＩＴＥ　ＯＮが必要だ。ＳＰＲＩＴＥ　ＯＮというのはスプライト表示モードのこと。スプライト面の表示を可能にする働きをもっているんだ。これを打ちこむことによって、スプライト面とバックグラウンド面が重ねて表示できる状態になり、スプライト面上にあるスプライト(アニメキャラクタ)が画面上に表示されることになるんだよ。

```
MOVE
```

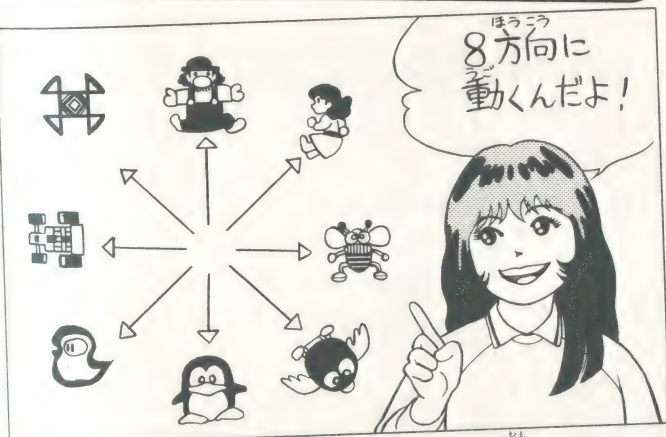
スプライトの座標は

ＳＰＲＩＴＥ　ＯＮで表示が可能になったアニメキャラクタはＤＥＦ　ＭＯＶＥ(49ページ)によって動かせるようになり、さらにＳＰＲＩＴＥによって画面に実際に表示される。画面に表示することのできるキャラクタの数は最高で16個。ただし水平方向に表示できるのは４個までだ。また、スプライト面の座標は水平方向(Ｘ)、垂直方向(Ｙ)とも0～255だけど、実際に有効な範囲はＸが0～240、Ｙが5～220までだ。

```
SPRITE ON
```

キャラクタの動きを指定する

アニメキャラクタの動きを指定する方法としてはDEF SPRITEを使う手もあるけど、ここでは、動作の方向を決めるだけで簡単にキャラクタを動かすことのできるDEF MOVEを紹介しよう。使い方は下の通りで、nはキャラクタの動作番号、Aはキャラクタの種類、Bは動作の方向指定、Cは動かす速さ、Dは全体移動量、Eは表示優先度、Fは配色番号だ。

```
DEFMOVE(N)=SPRITE(A,B,C,D,E,
F)
```

▲この命令を覚えればスプライトキャラは思いのままだ!

キャラクタを動かそう！

さあ、いよいよキャラクタを動かす段階だ。そのためにはMOVE文を使おう。nというのは、DEF MOVEであらかじめ定義したキャラクタの動作番号だ。n_0〜n_7まであって、同時には、8種類のキャラクタを画面上に表示させることができるぞ。ただし、画面の横方向に対しては最高4つまでしか表示できない。それ以上表示させようとしても消えてしまうから注意しよう。

```
10 SPRITE ON
20 DEF MOVE(0)=SPRITE(0,3,2,
10,0)
```

動かす最初の位置を決めよう！

キャラクタを動かす場合は、まず、その最初の位置をどこにするか決めなくちゃならないのだ。そのために必要なのがPOSITIONだ。

POSITIONによって決められる位置はもちろんスプライト面上。水平方向(X)、垂直方向(Y)とも0〜255の間ならどの位置にも指定できることにはなっているけど、スプライト面上で実際に有効な範囲はXが0〜240、Yが5〜220の間だ。

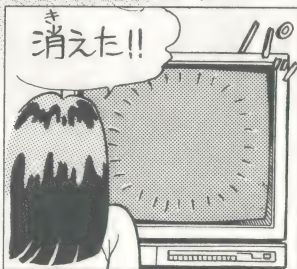
```
30 POSITION 0,60,120
40 MOVE 0
```

キャラクタを止まらせる

MOVE文でキャラクタが動きを開始することはすでに説明したよね。では、動き出したキャラクタを停止させるにはどうしたらいいんだろう。そんなときに使うのがCUTなんだ。n_0〜n_7の動作番号によって自分の好きなキャラクタの動きを止めることができるぞ。また同時に8個全部を停止させることも可能だ。動きを再開させるためにはMOVEを使う。停止位置から、残りの移動量を完了するまで動き始めるぞ。

```
10 CUT(N)
 :
 :
```

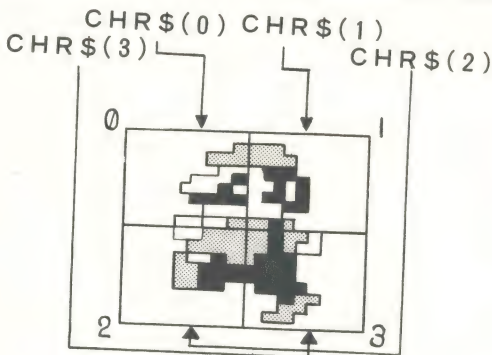
ERAでキャラクタを消す

MOVE文で動き始めたキャラクタを消したいときにはERAを使う。使い方はCUTとまったく同じ。CUTによって停止中のキャラクタも消すことが可能だ。MOVEで動きを再開させると消えた位置から再登場。

```
50 I$=INKEY$
60 IF I$=" " THEN ERA 0
70 IF I$<>CHR$(13) THEN 50
80 MOVE 0
```

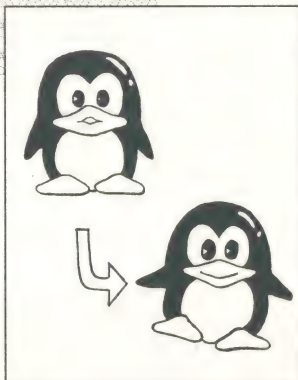
キャラクタの定義をしよう

アニメキャラクタをＳＰＲＩＴＥ命令によってスプライト面上の好きな位置に表示しよう。もし、水平方向、垂直方向の座標を省略して、ＳＰＲＩＴＥ　ｎだけを入力すると、表示されているアニメキャラクタの中から、指定のスプライト番号のキャラクタだけが消去される。

また、同一の座標に複数のアニメキャラクタを表示すると、スプライト番号の小さいものが手前に表示されるぞ。

```
20 DEF SPRITE 1,(0,1,0,0,0)=
CHR$(0)+CHR$(1)+CHR$(2)+CHR$
(3)
30 SPRITE ON:SPRITE 1,X,Y
```

▲スプライトを上手に使ってやればゲームも面白くなるゾ

スプライトの座標を読め！

動き出したキャラクタというのは、当然、その位置がどんどん変わってしまうよね。そこで、動作中のキャラクタの位置を調べるためにXPOS、YPOSというものがあるんだ。DEF MOVEで定義したキャラクタの動作番号を入れると、そのキャラクタが現在画面上のどの位置にいるか示してくれる。

```
LIST
5 CLS
10 SPRITE ON
20 DEF MOVE(0)=SPRITE(7,4,3,
70,0)
30 POSITION 0,20,30
40 MOVE 0
50 LOCATE 0,0
60 PRINT "X:";XPOS(0)
70 PRINT "Y:";YPOS(0)
80 IF MOVE(0)=0 THEN END
90 GOTO 50
```

Ｖ３のＶＣＴ命令

ファミリーベーシックＶ３にはＶＣＴという命令が追加されている。これは、ＤＥＦ　ＭＯＶＥによるアニメキャラクタの移動方向を読み取るときに使うものなんだ。

このＶＣＴ(ｎ)の値は０～８まで。アニメキャラクタが停止している場合は０になり、動いているときは１～８の数字でその方向を示す。１が真上、２が右上、３が右横、４が右下、５が真下、６が左下、７が左横、８が左上だ。

```
10 MOVE 0
20 PRINT VCT(0)
```

ＣＡＮでスプライトを消す

これもＶ３に追加された命令だ。いままでのファミリーベーシックでは、アニメキャラクタの表示を消すとともに、座標位置を未定義にするためにはＥＲＡ命令とＰＯＳＩＴＩＯＮ命令の２つが必要だった。だけど、Ｖ３では、このＣＡＮ命令だけで、ＤＥＦ　ＭＯＶＥで動きを定義したキャラクタを未定義にすることができるんだ。つまりＣＡＮというのはキャンセルのことなんだよ。

```
10 MOVE 0
20 CAN 0
```

スプライトの重なりを判定する

ＤＥＦ　ＭＯＶＥによるアニメキャラクタ同士の重なりというのはなかなか判定しにくいものだ。そこでＶ３では、この重なりを判定するために、ＣＲＡＳＨという命令が新たに追加された。

アニメキャラクタが重なっている場合は、ＣＲＡＳＨ(ｎ)は重なっているスプライトのうちでいちばん小さなスプライト番号となる。ただし重なっていない場合は－１、またスプライトが未定義の場合は－２となるんだ。

```
10 SPRITE ON
20 CGSET 1,1
30 DEF MOVE(0)=SPRITE(0,3,1,
255,0,0)
40 DEF MOVE(1)=SPRITE(0,7,1,
255,0,0)
50 MOVE 0,1
60 IF CRASH(0)<>-1 THEN PRIN
T "CRASH !!":BEEP
70 GOTO 60
```

アニメスプライトマップ

コード(10進数)	コード(16進数)	説明	コード(10進数)	コード(16進数)	説明	コード(10進数)	コード(16進数)	説明	コード(10進数)	コード(16進数)	説明
0	00	マリオ (WALK1)	32	20	レディ (WALK2)	64	40	アキレス (左1)	96	60	ペンペン (左歩1)
1	01		33	21		65	41		97	61	
2	02		34	22		66	42		98	62	
3	03		35	23		67	43		99	63	
4	04	マリオ (WALK2)	36	24	レディ (WALK3)	68	44	アキレス (左2)	100	64	ペンペン (左歩2)
5	05		37	25		69	45		101	65	
6	06		38	26		70	46		102	66	
7	07		39	27		71	47		103	67	
8	08	マリオ (WALK3)	40	28	レディ (JUMP)	72	48	アキレス (左上 1)	104	68	ペンペン (正面)
9	09		41	29		73	49		105	69	
10	0A		42	2A		74	4A		106	6A	
11	0B		43	2B		75	4B		107	6B	
12	0C	マリオ (JUMP)	44	2C	レディ (スリップ 着地)	76	4C	アキレス (左上 2)	108	6C	ペンペン (後)
13	0D		45	2D		77	4D		109	6D	
14	0E		46	2E		78	4E		110	6E	
15	0F		47	2F		79	4F		111	6F	
16	10	マリオ (スリップ 着地)	48	30	レディ (はしご)	80	40	アキレス (上1)	112	70	ファイアーボール(1)
17	11		49	31		81	51		113	71	
18	12		50	32		82	52		114	72	
19	13		51	33		83	53		115	73	
20	14	マリオ (はしご)	52	34	レディ (DOWN)	84	54	アキレス (上2)	116	74	ファイアーボール(2)
21	15		53	35		85	55		117	75	
22	16		54	36		86	56		118	76	
23	17		55	37		87	57		119	77	
24	18	マリオ (DOWN)	56	38	ファイターフライ(1)	88	58	ニタニタ (1)	120	78	車 (左1)
25	19		57	39		89	59		121	79	
26	1A		58	3A		90	5A		122	7A	
27	1B		59	3B		91	5B		123	7B	
28	1C	レディ (WALK1)	60	3C	ファイターフライ(2)	92	5C	ニタニタ (2)	124	7C	車 (左2)
29	1D		61	3D		93	5D		125	7D	
30	1E		62	3E		94	5E		126	7E	
31	1F		63	3F		95	5F		127	7F	

ゲームベーシック

一般的な命令

```
LIST
10 DATA "12345"
20 DATA "&HABC"
30 DATA "-123 "
40 DATA "+14  "
50 DATA "POS  "
60 FOR I=1 TO 5
70 READ A$,B$
80 PRINT A$ ; " --> " ; VA
A$)
90 NEXT
OK
RUN
12345 --> 12345
&HABC --> 2748
-123 --> -123
+14  --> 14
POS  --> 0
OK
```

これだけはおぼえておこう！
ゲームベーシックの基本命令だ

この項目で出てくる命令語

PRINT　INPUT　LINPUT　GOTO
GOSUB　RETURN　IF/THEN　END
FOR/TO/NEXT　ON/GOTO/GOSUB
READ　SWAP　STOP　PAUSE

画面に文字や数字を書くには？

ゲーム作りに欠かせない役割をはたすのがたし算、ひき算、かけ算、割り算などの計算だ。そしてこれらを画面表示させるときに使う命令がＰＲＩＮＴなんだ。

ＰＲＩＮＴによって出力される値は「；」か「，」で区切ることによって複数個をつづけて書くことができる。文字をすぐ後につづけて書く場合には「；」を、８文字間隔でつづけて書く場合には「，」を使おう。

```
A=120
OK
PRINT "A" , "=" ; A
A       = 120
OK
```

数値や文字を入力しよう！

キーボードから数値や文字を入力するときにはＩＮＰＵＴ文を使うんだ。このとき、ＩＮＰＵＴの後には必ず変数をつけること。というのは、データを入力する場合には、前もってデータを入れる変数を用意していなければならないからなんだ。また、文字だけの入力ならＬＩＮＰＵＴ文も使える。ＩＮＰＵＴでは「，」は「〃」で囲まなければ入力できないけど、ＬＩＮＰＵＴでは「，」もそのまま入力できるぞ。

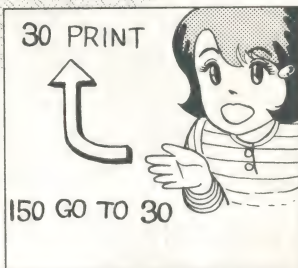
```
INPUT A$
?ABCDE

LINPUT B$
APL,SYS
```

GOTO文でジャンプ!

プログラムをキミの好きな行へいっきにジャンプさせたいときに使うのがGOTO文だ。GOTOの後に、ジャンプしたい行番号を指定すれば、プログラムはその行番号を実行してくれる。ようするにGOTO文というのは、指定した行番号に無条件にジャンプするステートメントなんだ。プログラムの流れを変えたいときに使用すると便利だよね。

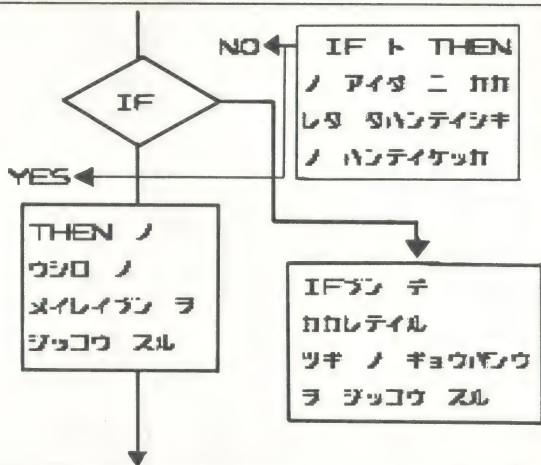
```
10 BEEP
20 FOR I=0 TO 800
30 NEXT
40 GOTO 10
```

GOSUB/RETURN

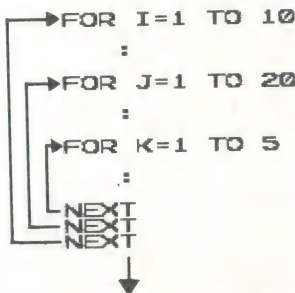
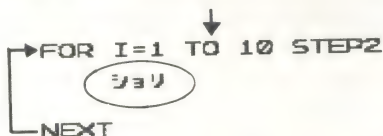
プログラム中、何回も繰り返して使うサブプログラムをサブルーチンという。このサブルーチンを呼びだすのがGOSUBだ。呼び出したサブルーチンの最後にはRETURNをつけて戻り先の行番号を指定しよう。

```
10 GOSUB 100
20 FOR I=0 TO 800:NEXT
30 GOTO 10
100 PLAY"ABC"
110 RETURN
```

ＩＦ－ＴＨＥＮで数値の判定を

ＩＦ文っていうのは条件分岐をおこなう命令で、ＴＨＥＮといっしょに使うんだ。使い方は、ＩＦとＴＨＥＮの間に判定式を書き、判定どおりになった場合にＴＨＥＮ以降を実行する。たとえば、ＩＦ　Ｘ＝10　ＴＨＥＮ　500というのは、Ｘが10であると判定されたら、行番号500へ飛んで、それを実行するということだ。もし判定されなければＩＦ文で書かれている次の行番号を実行するんだよ。

```
10 X=10
20 IF X=10 THEN BEEP
30 END
```

プログラムのループをおこなう

ループの間の処理を繰り返し実行するためには、FOR～TO／NEXTを使おう。この場合、FOR文はループの先頭を、NEXT文はループの終わりを示している。文法におけるループ変数、初期値、終了値、増分が指定できるぞ。ループは初期値から始まって、増分を加えながら処理を繰り返し、終了値を越えたところで終わりとなる。STEP命令を省略した場合は、増分が1になるんだよ。

```
10 FOR I=32 TO 255
20 PRINT CHR$(I) ; " ";
30 NEXT
```

READでデータを読み込む

データの入力にはＩＮＰＵＴ文のほかにＲＥＡＤ文による方法があるよ。ＩＮＰＵＴ文はプログラム実行時にデータを入力するけど、実行のたびに決まったデータを用いる場合には不便なので、こんなときはＲＥＡＤ文を使おう。

ＲＥＡＤ文はいつもＤＡＴＡ文と対にしていっしょに使うステートメントで、ＲＥＡＤの後に変数を書き、ＤＡＴＡの後にはそれに対応する定数データを書くんだよ。

```
READ
DATA
```

プログラム中にデータを書く

ＲＥＡＤ文で読み込むデータを用意するステートメントがＤＡＴＡ文だ。数値定数や文字定数のデータが用意でき、１行について255文字が入力可能だ。

ＤＡＴＡ文はプログラム中のどこにでも、いくつでも置くことができるけど、プログラムの読みやすさを考えた場合、ＲＥＡＤ文のすぐ後か、プログラムの最後にまとめて書くのがいい方法だよ。

```
10 DATA "ABC","123","POS"
20 FOR I=1 TO 3
30 READ A$
40 PRINT A$
50 NEXT
```

ON～命令

式の値によって、指定された行へジャンプする働きをするのがON～命令だ。

たとえば、式の値が1のとき、命令の後に書かれた行番号の並びの1番目にある行番号の行にジャンプするんだ。そして式の値が2ならば2番目、3なら3番目……というように対応していて、式の値が0のとき、あるいは指定した行番号の個数を越えるときはON文の次の文に移るんだよ。

```
10 INPUT "A=",A
20 ON A GOTO 100,200,300
30 PRINT "モウ 1 カイ"
40 GOTO 10
```

V3のON～命令

V3では、ON命令にON ERROR GOTOがくわわった。この命令はプログラム中にエラーが発生した場合、エラーの処理ルーチンに分岐させることが出来るんだ。ILエラーなどの処理にとてもいいんだ。

```
10 ON ERROR GOTO 100
20 ERROR 4
100 PRINT "ERROR !!"
110 END
```

SWAPで変数同士を交換

たとえば、A、Bという2つの変数があるとするよね。そこでいま、この2つの変数の内容を交換したいとする。こんなときいちいちプログラムを打ち直すのは大変だ。そこで役に立つのがSWAPなんだ。これを使えば簡単に、2つの変数の内容を交換することができるんだよ。ただし、交換する変数の型は一致していなければならないんだ。たとえばSWAP　A，B＄なんていうのはダメなんだよ。

```
10 A=10:B=30
20 PRINT A,B
30 SWAP A,B
40 PRINT A,B
```

プログラムを中止させるには？

プログラムの実行を止めるときにはSTOP命令を使うんだ。止まったプログラムは変数の中をクリアせず、CONT文によって次の文から再開できるので、プログラムの虫取りには最適のステートメントといえるね。

また、プログラムを一時休止させるときにはPAUSE命令、プログラムの終了を宣言するときにはEND命令を使おう。

```
10 PRINT "PAUSE"
20 PAUSE 200
30 PRINT "END"
```

ゲームベーシック

プログラムのための命令

```
K=STRIG(0)
IF K=1 THEN A$="ST
IF K=2 THEN
GOTO 10
```

これらの命令をおぼえておけば
プログラム作りも楽しくなるぞ

この項目で出てくる命令語

RUN	CLICK	FIND
LIST	AUTO	ERR
CONT	RENUM	ERL
NEW	DELETE	ERROR
FRE	TRON	

RUNでプログラム実行だ!!

プログラムを実行させるための命令がRUNだ。RUNを入力するとプログラムは最初から順に実行されていくぞ。ただし、このとき変数はすべてクリアされてしまうので気をつけよう。変数をクリアせずに実行したい場合は、GOTO(59ページ)、またはCONT(67ページ)を使おう。

また、プログラムを途中から実行させたい場合は、実行開始行番号をつけよう。

```
RUN
```

作ったプログラムを見る

自分で作ったプログラムを見直したいときにはLISTを使おう。この命令によって、メモリに入っているプログラムのうち、自分の見たい場所を画面に表示させることができるんだ。

また、ESCキーを押すと、表示が一時的に停止するからゆっくりとプログラムを見ることができて、ほかのキーを押すと、リストが再開されるぞ。

```
LIST
```

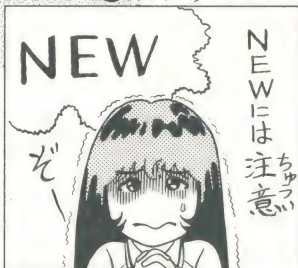
中止されたプログラムを再開！

プログラム上のSTOP命令によってプログラムが停止している場合や、STOPキーを押してプログラムが停止している場合に、CONTを入力すると、プログラムは停まった次の行番号の命令文から実行を開始するぞ。この場合、停止する前までの変数は保持されたままだ。ただし、ENDやエラーの発生によりプログラムが停まったときや、CLEAR実行直後などはCONTは正しく実行されないので注意！

```
CONT
END
OK
```

NEWはプログラムを消す！

新しいプログラムを入れるときは、それまでメモリに入っていた古いプログラムを消しておいたほうがトラブルがない。こんなときNEWを入れれば古いプログラムをすべて消すことができるんだ。

```
NEW
OK
LIST
OK
```

FREで残りのメモリエリアを

ユーザーメモリの未使用領域のサイズを与える働きをもっているのがＦＲＥだ。

ＦＲＥによって示される関数の値は、ベーシックのプログラムで使っていないユーザーメモリのバイト数。

ただし、ベーシックのバージョン、変数の状態、プログラムの有無、プログラムの実行前後によってその値は違ってくるから気をつけよう。

```
10 PRINT "ABCDEFG"
PRINT FRE
 1963
```

CLICKでキーの音を消す

ＣＬＩＣＫというのはＶ３に新しく追加された命令だ。

このＣＬＩＣＫによって、キーの入力音(クリック音)を消したり、発生させたりすることができるんだよ。

つまり、ＣＬＩＣＫ　ＯＮでクリック音が発生し、ＣＬＩＣＫ　ＯＦＦでクリック音が消えて無音になるというわけ。

キミの好みや、そのときの状況に応じて使いわけるといいだろうね。

```
CLICKON
CLICKOFF
```

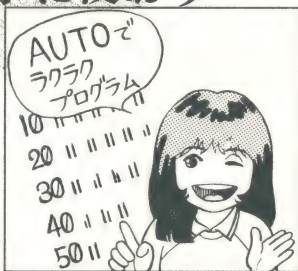
自動的に行番号を出力！

Ｖ３に新たに追加されたＡＵＴＯ命令を使うと、これから入力するプログラムの始めの行番号(ｍ)と、いくつずつ増加させて行番号を付けるのか(ｎ)、その値を指定することによって自動的に行番号を付けることができるぞ。

したがって、入力するのは、自動的についた行番号のあとのプログラム部分だけ。RETURNキーを押すと、指定した増加分で次の行番号が付けられ表示されていくんだ。

```
AUTO  M , N
```

ＡＵＴＯを上手に使おう

ＡＵＴＯ命令で、プログラム中にすでにある行番号が表示された場合、前のプログラムを残すときは、RETURNキーを押してその行を飛ばす。文字を入力してからRETURNキーを押すと行の内容は新しくなる。

```
AUTO 10
OK
10
20
30
```

プログラムをきれいに並べよう

プログラムの行番号を整理する命令もＶ３にも追加されている。この命令はＲＥＮＵＭといって、これを使うと行番号が整理されるだけでなく、ＧＯＴＯ、ＧＯＳＵＢ文などの分岐先行番号も新しい行番号に対応して自動的に変更されるんだ。ただし、ＥＲＬ＝1000のように、定数で指定した行番号は変更されないから注意しよう。また、文法中のＬは新しくつける開始行番号のこと。ｍとｎはＡＵＴＯと同じだ。

```
RENUM L,M,N
```

ＤＥＬＥＴＥでプログラム消去

ＤＥＬＥＴＥもＶ３の新命令。プログラムの中に消したい部分があるとき、その行番号を指定することによって、まとめて取り消すことができるようになったぞ。ｍは取り消しを始める行番号で、ｎは取り消しを終える行番号だ。ｍの値だけを指定した場合は、その行だけが取り消され、一からｎの値だけを指定した場合はプログラムの先頭から指定行(ｎ)までが取り消される。ｎを省略すると、ｍから最終行まで消去。

```
DELETE M - N
```

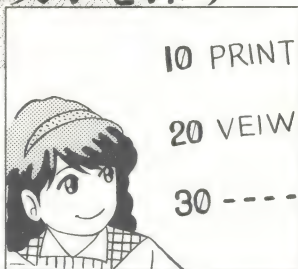
TRONでプログラムトレース

TRONもV3の新しい命令だ。TRON文により、トレースモードを実行し、プログラム実行中の行番号を表示させることができるぞ。ちなみにTRON文の実行はダイレクトモードでもプログラムモードでも可能だ。

また、TROFFにすると、TRON命令は終了し、ノーマルモードに戻る。2つのトレース命令の使い方をマスターしよう。

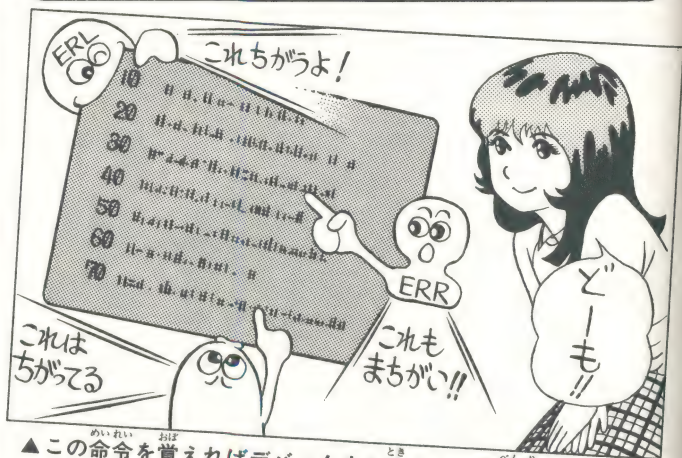
```
TRON
TROFF■
```

プログラム中の文字を探す

プログラムの中から指定した文字列(X)を探し出し、それをふくむ行を表示させる命令がV3のFINDだ。プログラムを変更したいときなどにこの命令を使うととても便利だぞ。文字列は31字以内。

```
FIND "X"■
```

エラーの種類をたしかめよう

V３にはエラーに関する命令が３つ追加されている。ひとつは、エラーが発生したときそのエラーコードを求めることができるＥＲＲ。ふたつめは、エラーが発生した行番号を求められるＥＲＬ。この２つは、エラーが発生したとき、その原因がどこにあるのかすぐに知ることができる便利な命令だ。そしてもうひとつは、仮のエラーを発生させるＥＲＲＯＲ。エラー処理ルーチンが正しく働くかどうかを確認する命令だ。

```
10 ON ERROR GOTO 30
20 ERROR 4
30 PRINT ERL
40 PRINT ERR
50 END
```

▲この命令を覚えればデバックする時にとても便利だよ

ゲームベーシック

画面制御命令

```
KEY LIST REN
F1  GAME 0(M)
F2  GAME 1(M)
F3  GAME 2,1(M)
F4  GAME 3(M)
F5  SPRITE
F6  LOAD(M)
F7  LIST
OK
```

キャラクタに色を着けて
カラフルな画面を作り出そう！

この項目で出てくる命令語

PRINT　LOCATE　VIEW　CLS
CGEN　COLOR　CGSET　FILTER

PRINTする座標を決める

ふつう、入力されたプログラムは画面の上から下へと表示されるよね。でも、LOCATEという命令を使えば、文字や数字を自分の好きな場所に移動させることができるんだ。

LOCATEによるカーソルの位置指定は水平方向の表示カラム(X)と垂直方向の表示行(Y)によっておこなう。Xの範囲は0～27、Yの範囲は0～23だ。この範囲内ならば、自由にカーソルを移動させることができるんだよ。

```
10 CLS
20 LOCATE 0,0
30 PRINT "ABC"
40 LOCATE 5,5
50 PRINT "POS"
```

BGグラフィックの合成には？

BGグラフィック面をバックグラウンド面へコピーするときは、VIEW命令を使おう。ベーシックを実行中、VIEW命令を入力すると、BGグラフィック面に描いた絵が、バックグラウンド面へコピーされるぞ。このとき、BGグラフィック面に描いた絵をそのままの色でバックグラウンド面に表示するためには、バックグラウンド面のパレットコードは1(CGSET 1, 1)を使用しよう。

```
VIEW
```

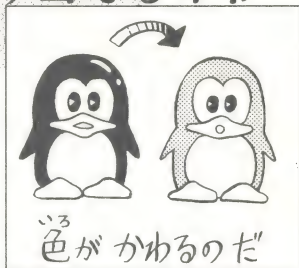
CLS命令で画面をクリア

画面をクリアしたいときには、CLS命令を使おう。これによって、バックグラウンド面がクリアされるぞ。このときバックグラウンド面にコピーされたBGグラフィックも同時に消える。プログラム上でBGグラフィックをバックグラウンド面にコピーする場合は、CLS命令と入れ替えてVIEW命令を使用しよう。また、クリアされた画面の左上にはOK表示が出るぞ。

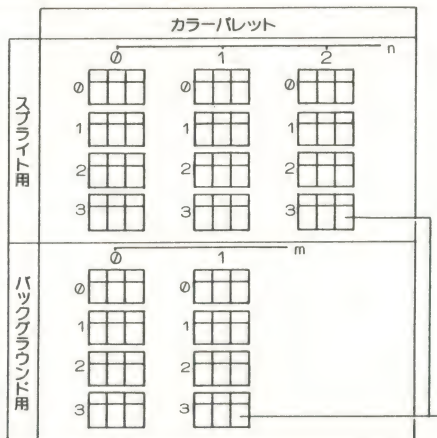
```
CLS
```

キャラクタを割り当てる命令

バックグラウンド面およびスプライト面にキャラクタの割り当てを決めるときに使うのがCGEN命令だ。バックグラウンド面にアニメキャラクタを表示したり、スプライト面に文字記号を表示することができるぞ。

```
CGEN 2
```

COLORで画面に色を

バックグラウンド面へ表示する文字の配色番号を画面上のエリア毎に指定するのがCOLORだ。水平方向の表示カラムは0～27、垂直方向の表示行は0～23の範囲でおこなうことができ、配色番号は0～3まで。X、Yで指定された位置をふくむ画面上のエリア毎に、背景、または文字の表示色をCGSETで指定されたカラーパレットコード内の配色番号の中から選ぶんだ。

```
20 PRINT "[illegible]"
30 COLOR 0,0,0
40 COLOR 2,0,1
50 COLOR 4,0,2
60 COLOR 6,0,3
```

パレットコードを指定する

BGやスプライトで使用するパレットの割り当てを決めるときに使うのがCGSETだ。用意されている色の組み合わせのグループの中から好きな色を選び、背景の色やアニメキャラクタの表示色を決めよう。

準備されているカラーパレットはバックグラウンド用で2種類、スプライト用で3種類。それぞれのカラーパレットには計12色の色コードが保持できるぞ。

```
CGSET 0,1
```

《52色の色コード》

色相	16進	10進	16進	10進	16進	10進	16進	10進	
	00	0	10	16	20	32	30	48	灰色~白色
青系	1	1	11	17	21	33	31	49	有色
青系	2	2	12	18	22	34	32	50	有色
青系	3	3	13	19	23	35	33	51	有色
青系	4	4	14	20	24	36	34	52	有色
赤系	5	5	15	21	25	37	35	53	有色
赤系	6	6	16	22	26	38	36	54	有色
赤系	7	7	17	23	27	39	37	55	有色
赤系	8	8	18	24	28	40	38	56	有色
緑系	9	9	19	25	29	41	39	57	有色
緑系	A	10	1A	26	2A	42	3A	58	有色
緑系	B	11	1B	27	2B	43	3B	59	有色
緑系	C	12	1C	28	2C	44	3C	60	有色
	0D	13	1D	29	2D	45			クロ
	0E	14	1E	30	2E	46			クロ
	0F	15	1F	31	2F	47			クロ

暗←→明

SCREENで画面のきりかえ

SCREENはV3の新しい命令だ。BG面を2つもっているので、表示面とアクティブ面をそれぞれ別々に指定できるという長所があるぞ。また、表示面とアクティブ面が同一の場合は、アクティブ面の指定を省略することもできるんだ。ちなみにアクティブ面っていうのは、カーソルが表示されており、カーソルを動かして実際に表示を変更できる面のことをいうんだよ。

```
20 DEF SPRITE 0,(3,1,0,0,0)=
CHR$(0)+CHR$(1)+CHR$(2)+CHR$
(3)
25 PALETS 3,&HF,&H30,&H30,&H
30
30 SPRITE 0,100,150
```

バックカラーの指定もできるぞ

BG面の全面着色ができる命令がV3のFILTERだ。これをおこなうと、全面がフィルターをかけたように淡く着色されるんだ。

色の指定番号は、0〜7の8種類。なにも番号を指定しない場合はデフォルト値として0(無色)が選択される。また、CTRキーとDキーを同時に押してもデフォルト状態になるぞ。

```
SCREEN 0,0
```

キャラクタに好きな色を

キャラクタに着色したいときはPALETを使おう。背景(B)とアニメキャラクタ(S)を52色の色コードの中から好きな色を選んで着色できるぞ。また、バックドロップ面の色、はバックグラウンドで使用しても、スプライトで使用しても画面に表示着色されるんだ。

キミの作ったキャラクタや背景に色を着けて、ゲームをカラフルなものにしよう。

```
FILTER 1
```

FILTER カラー番号

Ø番	1番	2番	3番	4番	5番	6番	7番
無色	赤	緑	黄	青	マゼンタ	空色	白

ただし，デフォルト値はØ

▲この中から好きな色を指定出来る。カラーページの表も見よう

10進数と16進数のこと

スプライトやキャラクタ表などにA3とかBFなど数字とも文字とも見わけられないのがあるだろう。これは16進数と言って私達が使っている数字とはちがうものなんだ。

10シン	16シン	10シン	16シン	10シン	16シン	10シン	16シン	10シン	16シン	10シン	16シン
0	0	16	10	32	20	48	30	64	40	80	50
1	1	17	11	33	21	49	31	65	41	81	51
2	2	18	12	34	22	50	32	66	42	82	52
3	3	19	13	35	23	51	33	67	43	83	53
4	4	20	14	36	24	52	34	68	44	84	54
5	5	21	15	37	25	53	35	69	45	85	55
6	6	22	16	38	26	54	36	70	46	86	56
7	7	23	17	39	27	55	37	71	47	87	57
8	8	24	18	40	28	56	38	72	48	88	58
9	9	25	19	41	29	57	39	73	49	89	59
10	A	26	1A	42	2A	58	3A	74	4A	90	5A
11	B	27	1B	43	2B	59	3B	75	4B	91	5B
12	C	28	1C	44	2C	60	3C	76	4C	92	5C
13	D	29	1D	45	2D	61	3D	77	4D	93	5D
14	E	30	1E	46	2E	62	3E	78	4E	94	5E
15	F	31	1F	47	2F	63	3F	79	4F	95	5F

10シン	16シン	10シン	16シン	10シン	16シン	10シン	16シン	10シン	16シン	10シン	16シン
96	60	112	70	128	80	144	90	160	A0	176	B0
97	61	113	71	129	81	145	91	161	A1	177	B1
98	62	114	72	130	82	146	92	162	A2	178	B2
99	63	115	73	131	83	147	93	163	A3	179	B3
100	64	116	74	132	84	148	94	164	A4	180	B4
101	65	117	75	133	85	149	95	165	A5	181	B5
102	66	118	76	134	86	150	96	166	A6	182	B6
103	67	119	77	135	87	151	97	167	A7	183	B7
104	68	120	78	136	88	152	98	168	A8	184	B8
105	69	121	79	137	89	153	99	169	A9	185	B9
106	6A	122	7A	138	8A	154	9A	170	AA	186	BA
107	6B	123	7B	139	8B	155	9B	171	AB	187	BB
108	6C	124	7C	140	8C	156	9C	172	AC	188	BC
109	6D	125	7D	141	8D	157	9D	173	AD	189	BD
110	6E	126	7E	142	8E	158	9E	174	AE	190	BE
111	6F	127	7F	143	8F	159	9F	175	AF	191	BF

10シン	16シン	10シン	16シン	10シン	16シン	10シン	16シン
192	C0	208	D0	224	E0	240	F0
193	C1	209	D1	225	E1	241	F1
194	C2	210	D2	226	E2	242	F2
195	C3	211	D3	227	E3	243	F3
196	C4	212	D4	228	E4	244	F4
197	C5	213	D5	229	E5	245	F5
198	C6	214	D6	230	E6	246	F6
199	C7	215	D7	231	E7	247	F7
200	C8	216	D8	232	E8	248	F8
201	C9	217	D9	233	E9	249	F9
202	CA	218	DA	234	EA	250	FA
203	CB	219	DB	235	EB	251	FB
204	CC	220	DC	236	EC	252	FC
205	CD	221	DD	237	ED	253	FD
206	CE	222	DE	238	EE	254	FE
207	CF	223	DF	239	EF	255	FF

ゲームベーシック

音楽用命令

コンピュータは楽器にもなる！
効果音でゲームを盛りあげよう

この項目で出てくる命令語

PLAY　BEEP

PLAY命令で音楽を

PLAY命令を使えば、音楽を演奏させることができるぞ。そのためには、音楽用のデータを指定しなければならない。テンポ(T)は速い1から遅い8まで、音色(Y)は4種類が用意されている。また、エンベロープをかけたいときはM1、かけないときはM0を使用。音量はVで表わし、0～15の範囲で大小が決まる。オクターブの指定は0によって指定、0が低音で5が高音だ。音程はC～Bの文字で表わすんだ。

```
LIST
10 PLAY "V15M1Y103"
20 PLAY "CEDFAGB"
OK
```

エンベロープ…M0、M1でエンベロープをかけるか否かを指定します。エンベロープがかかると、余いんのある音、切れた音、はねた音になります。

●M0(エンベロープをかけない)のとき、VnでV0～V15で音量を指定します。

0 ←小―音量―大→ 15

M0V15 ―― 音量15となる

●M1(エンベロープをかける)のとき、VnはV0～V15でエンベロープの長さを指定します。

0 ←短い―エンベロープ―長い→ 15

M1V3 ―― エンベロープ3となる。音量はV15の音量となります。

任天堂マニュアルを転載しました。

BEEPでエラーの音を出す

ゲームをやってても なんにも音がしないのではつまらないよね。やっぱり、失敗したときには「ピー」という音が鳴ってくれたほうが、ゲームにケジメがつくっていうもんだ。

この「ピー」という音を出すためにはBEEP命令を使うんだ。プログラムにBEEPを入力しておけば、ゲームの失敗ごとに音が鳴る。これでキミのゲームもかなりそれっぽくなってくるぞ。

```
10 BEEP
```

音　　程	指定方法
ド	C
ド#(レb)	#C
レ	D
レ#(ミb)	#D
ミ	E
ファ	F
ファ#(ソb)	#F
ソ	G
ソ#(ラb)	#G
ラ	A
ラ#(シb)	#A
シ	B

ベーシックはこうなっている

PEEKやPOKEなどを使うと～番地なんて出てくるだろう。これはベーシックのメモリのアドレスのことだ。V3は、メモリマップを見るとV2との違いはひと目でわかるよ。

(16進表現)	
&H0000 &H07FF	ワークRAM（ファミリーコンピュータ本体内）
&H0800 &H1FFF	未使用
&H2000 &H5FFF	システムで使用
&H6000 &H6FFF	未使用
&H7000 &H703F	
&H77FF	ワークRAM（BASICカセット内）用 各種ボードデータまたはBASICフリーエリア
&H7800 &H7FFF	未使用
&H8000 &HFFFF	プログラムROM

ゲームベーシック

関数(かんすう)

関数(かんすう)を使(つか)いこなせるようになれば
ベーシックがより楽(たの)しくなるぞ

この項目(こうもく)で出(で)てくる命令語(めいれいご)

ABS	VAL	MID$
SGN	HEX$	LEN
RND	RIGHT$	INSTR
ASC	LEFT$	CHR$

絶対値と数値の符号

まず、数式の絶対値を出すためにはＡＢＳ文を使う。絶対値というのはプラスマイナスに関係なくその数字を数えたときの数のことだ。たとえば10も－10も絶対値は10ということになる。

つぎに、数式の符号を求めるときにはＳＧＮ文を使う。これは、その数式が０より大きければ１、０ならば０、０より小さければ－１というふうに示されるんだ。

```
10 A=10:B=-20
20 PRINT A,B
30 A=ABS(A):B=ABS(B)
40 PRINT A,B
```

ＲＮＤで乱数を発生させよう

ＲＮＤ命令は乱数と言って指定した数値以内の数字をランダムに出してくれる関数なんだ。リアルタイムのゲームなどに、ペンペンやニタニタなどのスプライトキャラクタを敵として出している場合が多いだろう。そのほとんどが、ＲＮＤ命令を使った動きをしているぞ。ＲＮＤの良い所は人間につぎに出てくる数字を予想させないので数あてゲーム的なことも簡単に出来るんだ。ＲＮＤは必ずマスターしよう。

```
10 X=RND(26)
20 Y=RND(22)
30 LOCATE X,Y
40 PRINT "A"
50 GOTO 10
```

ASCで文字を数値に交換

文字コードを数値に変換するときには、ASCを使う。ABSの後に文字列を書くと、その左端の文字を数値に変換するぞ。つまり、文字列の最初の1文字のキャラクタコードがこの関教の値となるわけだ。

キャラクタコードの値は0～255の整数値。文字列は式や変数でもかまわない。また、文字列がヌルストリングのときは、この関数の値は0になるぞ。

```
10 A$="C"
20 PRINT "A$=";A$
30 PRINT "ASC(A$)=";ASC(A$)
```

数式の値を文字に変換する

数値をキャラクタコードとみなして、対応する文字に変換する。つまりASCと逆の働きをするのがCHR$だ。この場合は数値1つにつき1文字が得られる。プリント表示される文字、記号はXの値が32から255だ。

```
10 A=10:B=-20
20 PRINT A,B
30 A=SGN(A):B=SGN(B)
40 PRINT A,B
```

数字の文字列を数値に変換

文字列の中の文字としての数字を数値に変換する働きをするのがＶＡＬだ。数字の文字列の範囲は、－32768～＋32767、＆Ｈ０～＆Ｈ７ＦＦＦ。もし最初の文字が、＋、－、＆、または数字でない場合は、この関数の値は０になる。

また、文字列の中に数字以外の文字（16進数の場合はＡ～Ｆをのぞく）が現われたら、それ以降の文字は無視するしくみになっている。

```
10 DATA "1234"
20 DATA "&HABC"
30 DATA "-123"
40 FOR I=1 TO 3:READ A$
50 PRINT VAL(A$):NEXT
```

数式を16進数の文字列に変換

ふだん、キミたちが日常生活の中でよく使っている数字は10進法にもとづいている。つまり、10倍ごとにケタをひとつあげる数え方だよね。ところがファミリーベーシックでは10進法以外に、16進法という数え方をよく使用するんだ。この16数法による数が16進数。そこで、数式を16進数の文字列に変換するためのステートメントも用意されているわけで、これがＨＥＸ＄なんだ。

```
10 INPUT "10 シン";A
20 PRINT "16 シン";HEX$(A)
30 PRINT
40 GOTO 10
```

文字列から指定した文字を取る

文字列の左側から、指定した数だけ文字を取り出す働きをするのがＬＥＦＴ＄、逆に、文字列の右側から指定した数だけ文字を取り出す働きをするのがＲＩＧＨＴ＄だ。たとえば〝コンニチハ〟という文字列から２文字を取り出すときＬＥＦＴ＄を使えば、〝コン〟になり、ＲＩＧＨＴ＄を使えば、〝チハ〟になる。この取り出す文字数(ｎ)が文字列の文字数より大きいときは、すべての文字が関数の値となるよ。

```
10 A$="コンニチハ"
20 PRINT RIGHT$(A$,2)
30 PRINT LEFT$(A$,2)
```

文字を指定した数だけ取る

文字列の中の好きな位置から文字を取り出すときにはＭＩＤ＄を使う。このときは、文字列の最初の文字を１とみなすから、〝コンニチハ〟の中から〝ンニ〟を取り出したい場合は、開始位置を２にすればいいんだ。

```
10 A$="コンニチハ"
20 PRINT MID$(A$,2,3)
```

LENで文字の長さを求める

LENというのは、文字列の文字数を与える働きをするステートメントだ。つまり、この関数の値は、文字列にふくまれるすべての文字数ということになる。

文字の数は0～31で、文字列がヌルストリングのときは0になる。また、空白やコントロールコードなどの画面表示されない文字も1文字とみなして数えられるから、気をつけておこう。

```
10 INPUT "ナマエ ハ " ; A$
20 PRINT "ナガサ ハ " ;
30 PRINT LEN(A$) ;
40 PRINT " モジ デス"
```

文字の位置を求める

指定した文字列の中から指定した文字の位置を求められる命令がV3には追加されている。これがINSTRで、これを使う場合、指定した文字列を文字列1と呼び、指定した文字を文字列2と呼ぶ。つまり、文字列1の中から文字列2の位置が探し出せるというわけだ。たとえば、〝ABCEF〟の中から〝A〟を探すと、1という値が与えられる。最初から1文字目に〝A〟が位置しているというわけだ。

```
PRINT INSTR("ABCEF","A")
```

ゲームベーシック

特殊な関数

この項目で出てくる命令語

POS　CSRLIN　STICK　STRIG
INKEY$　SCR$

POSでカーソルの水平値を

現在、画面上のカーソルの水平位置がどこにあるのかを知りたいときにはPOSを使用するんだ。

ゲームなどにはあまり使われていないんだけれどもLOCATE命令で指定したX座標を読み取ることが出来るのでテキスト画面を使ったゲームに使って見るのもいいだろう。

カーソルの水平値はあまり使うことはないけど、XPOSなどのグラフィック座標を読む命令には良く使うんだ。

```
10 LOCATE 10,10
20 X=POS(0)
30 LOCATE 0,13
40 PRINT X
```

カーソルの垂直値を求める

CSRLIN命令はカーソルの垂直値を読み取る命令なんだ。この命令はLOCATE命令を使わないPRINT命令だけの時に使う命令だけど、POSとならんでBASICではあまり使われない命令だ。古いタイプのゲームでテキストゲームと言うのがあって、グラフィックやLOCATEをあまり使わず、プリンターを画面のかわりに使っていたゲームに良く使われていたんだ。

```
10 LOCATE 10,10
20 Y=CSRLIN
30 LOCATE 0,13
40 PRINT Y
```

押されたキーを読み取る

キーボードから入力された文字がなにかを読み取る働きをするのがＩＮＫＥＹ＄だ。つまり、この関数の値は、キーボードから入力された１文字ということになるんだ。文法中の(ｎ)は引数で、これが０の場合は、カーソルをブリンクし、１文字の入力があるまで待つ。また、引数が省略された場合で、キーが押されているときは、その文字が関数の値になり、押されていないときはヌルストリングが関数の値になるよ。

```
10 I$=INKEY$
20 IF I$="" THEN 10
30 BEEP
40 GOTO 10
```

ＢＧグラフィック面のＳＣＲ＄

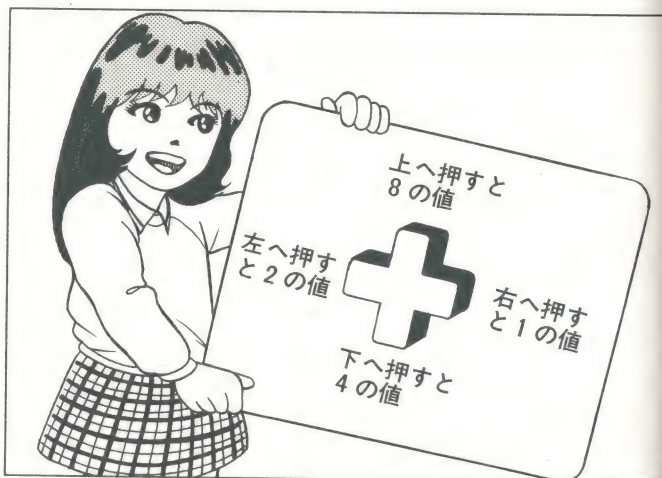
ＳＣＲ＄は、ＢＧグラフィック画面で表示されている文字、または絵を求める関数だ。ＢＧグラフィック画面上のカラムと行を指定すると、そこに書かれている文字や絵、配色番号を求めることができる。

```
10 CLS
20 PRINT "FAMILY BASIC"
30 PRINT SCR$(0,0)
```

STICKで方向を読み取ろう

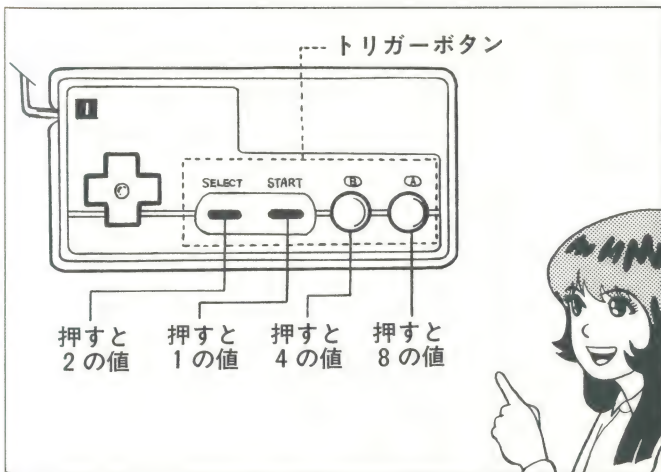
コントローラの✚ボタンが現在どの方向を示しているのかを読み取るにはSTICK文を使用する。STICKっていうのは、コントローラの✚ボタンからの入力値を求める働きをするものなんだ。

STICKの後の指定には0か1が入る。0はⅠコントローラ、1はⅡコントローラに対応している。対応する値は、1が右、2が左、4が下、8が上だ。

```
10 K=STICK(0)
20 PRINT K
30 GOTO 10
```

ボタンの情報を読み取れ！

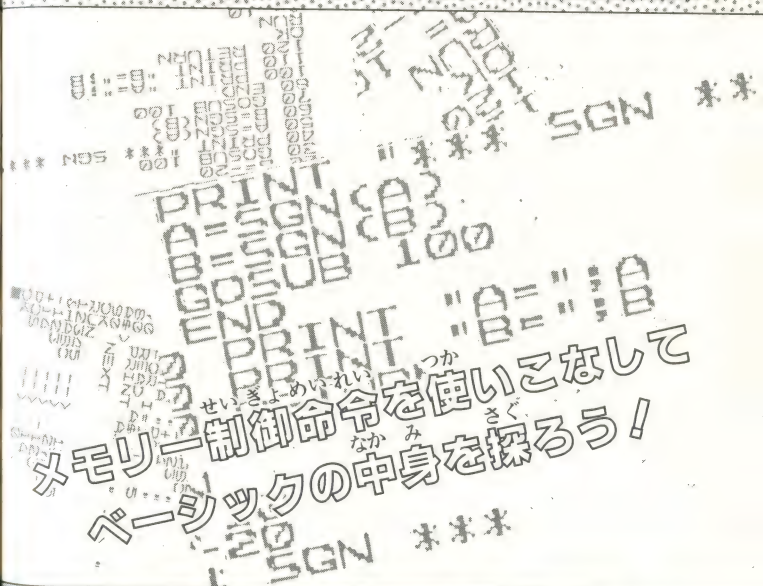
コントローラのトリガーボタンからの入力状態値を求めるときに使うのがＳＴＲＩＧだ。つまりＳＴＲＩＧ文を使えば、現在、どのボタンが押されているのか、すぐにわかるんだ。

Ⅰコントローラのボタン情報を得るためには０を、Ⅱコントローラの場合は１を指定する。ボタン情報の値は、スタートボタンが１、セレクトボタンが２、Ⓑボタンが４、Ⓐボタンが８だ。

```
10 K=STRIG(0)
20 PRINT K
30 GOTO 10
```

BASICの仕様水準

文字の種類	数字・英文字・カナ・記号
数の表現範囲 (整数)	10進 (-32768~+32767) 16進 (&HØØØØ~&HFFFF) 文字列 (Ø~31文字)
変数名の種類	英字で始まる先頭2文字を識別。長さ255桁
行番号の範囲	Ø~65535
テキスト行の最大桁	255桁
配列の大きさ	2次元まで要素数制限なし (メモリの範囲内)
マルチステートメント	可能、:(コロン) で区切る
サブルーチン、ネスト	制限なし (メモリの範囲内)
FOR NEXTループ	制限なし (メモリの範囲内)
編集機能	スクリーンエディタ
画面モード	BGグラフィック面、スプライト面、バックグラウンド面 バックドロップ面
グラフィック精度	バックグラウンド面 (28文字×24行) スプライト面 (256×24Øドット) 1キャラクタ (8×8ドット)
カラー	カラーゼネレータ52色発生
フィルター機能	カラーØ~7の着色可能
サウンド機能	音階、テンポ、3重音、音色
マイク音入力機能	音入力の有無検出
コントローラ入力	ⅠⅡコントローラ (方向入力、トリガー入力可能)
ファイル機能	カセットテープ (1,2ØØボー)
命令数	基本1ØØ
アニメキャラクタ設定	既設16種類より選択

ゲームベーシック

メモリー制御命令

この項目で出てくる命令語

PEEK　POKE　CALL　CLEAR

PEEKでメモリを読み取る

指定したメモリアドレスからデータを取り出すときに使うのがPEEKだ。

まず、PEEKと入力してから、その後に、取り出したいメモリ内のアドレスを指定する。

これによって、メモリ内の指定したアドレスから取り出した1バイトのデータがこの関数の値になるんだ。

次のPOKEとは反対の働きをするわけだね。

```
10 PRINT PEEK(&H7200)
```

POKEでメモリを書き込む

メモリ上の指定したアドレスに、直接1バイト(8ビット)のデータを書き込むときに使うのがPOKEだ。このとき、データは、0～255の値でなければいけない。データを「，」で区切って書くと、アドレスで指定したアドレスから連続したアドレスに書き込むことができるぞ。ただし、POKEは現在のメモリ内容を書き換えてしまうので、不用意に使うとファミリーベーシックのシステム領域を壊すから注意！

```
10 POKE &H7200,&HFF
```

CALLでマシン語にジャンプ

アドレスを実行開始番地とするマシン語サブルーチンを直接呼び出す働きをするのがCALLだ。アドレスには、16進法(&H○○○○の形)、10進整数定数(−32768〜+32767)などの整数定数、整数変数、式が指定されるんだ。

マシン語サブルーチンは、CLEAR文(99ページ)で指定したアドレス以降に置くようにする。ちなみにこのCALLの省略形はCA.だ。

```
CALL &H7000
```

メモリエリアの指定

メモリ内でBASICが使用する領域の最上位アドレスを指定するためのものがCLEAR文だ。この命令はふつう、プログラムの先頭部分に書かれ、サブルーチンの中では使用できないことをおぼえておこう。

```
CLEAR &H7600
```

PEEK, POKEの使い方

PEEK、POKE、CALLなどの命令は、あることは知っていても実際、使っている人は少ないんじゃないのかな。又、ベーシックの中はどうなっているか知ってみたいと思っている人もいると思うんだ。下のプログラムは、16進でメモリに書き込むことが出来るプログラムで書き込むアドレスを、16進で入力してからアドレスが出てきたらデータを入力してやるんだ。その下はチェック用のダンプリストだ。

```
10 INPUT A
20 POKE &H7200,A
30 A=PEEK(&H7200)
40 PRINT A
```

```
LIST
10 INPUT "ADDR:",A$
20 A=VAL("&H"+A$)
30 FOR I=0 TO 7
40 PRINT RIGHT$("000"+HEX$(A
),4);
50 FOR J=0 TO 7
60 PRINT " ";RIGHT$("0"+HEX$
(PEEK(A)),2);
70 A=A+1
80 NEXT
90 NEXT
100 INPUT "",A$
120 GOTO 30
OK
■
```

ゲームベーシック

SAVE・LOAD

この項目で出てくる命令語

LOAD　SAVE　BGGET　BACKUP

Ｖ２でバックアップする

入力したプログラムを消さないで保存しておくための方法がバックアップだ。まずＬＩＳＴでプログラムを確認し、次にＳＹＳＴＥＭと入力後、RETURNキーを押す。するとゲームベーシックモード画面になるので、３のＥＮＤを選ぶ。そしてここでバックアップスイッチをＯＮにすれば保存完了。

```
SYSTEM
```

ＳＡＶＥでプログラムを保存

バックアップのほかにプログラムを保存する方法としてＳＡＶＥがある。これはプログラムをカセットテープに記録する方法で、ＳＡＶＥ命令によっておこなうんだ。このときファイル名(プログラムの名前)を入れるのを忘れないように。下の文法２はＢＧグラフィックのＳＡＶＥに使用する。

1
```
SAVE
```

2
```
SAVE "ABC
```

SAVEでプログラムを出力!

SAVEをおこなうにはカセットテープレコーダが必要だ。キーボードのWRITE端子とテープレコーダのマイク端子、キーボードのREAD端子とテープレコーダのイヤホン端子をそれぞれ接続して準備完了。そして前ページのようにSAVE命令をおこない、テープレコーダの録音ボタンを押す。テープが回り始めたら RETURN キーを押す。しばらくすると画面にOKと表示されカーソルがでる。これで記録完了。

```
SAVE
```

バックアップから読み込む

バックアップによって保存したプログラムをこんどは復元させてみよう。

まず、コンピュータの電源を入れる。するとスタート画面が表示されるので、なにかキーを押す。「バックアップスイッチをOFFにしてください」と表示されたらその通りにする。これで「BASICのデータが残っています」と表示されたら復元したということだ。LISTで確かめてみよう。

```
BACKUP
```

LOADでプログラムを入力！

SAVEの逆がLOAD、つまり、カセットテープに記録されているプログラムを読み込んでメモリに記録することだ。
まず、カセットテープを読み込みたいプログラムの入っている頭の部分まで巻きもどしておき、データレコーダのLOAD端子とキーボードのREAD端子にピンジャックをつなぐ。

```
LOAD
```

ファイル名を書き、LOAD！

テープレコーダの準備が完了したら、キーボードでLOAD命令をする。そしてその後にファイル名を指定して、RETURNキーを押す。これであとはテープレコーダの再生ボタンを押せばいいってわけ。
BGグラフイックは、ベーシックでLOAD出来ないんだ。

1
```
LOADS
```

2
```
LOAD?
```

Ｖ３でのＢＧグラフィックＳＡＶＥ

ファミリーベーシックＶ３ではこれまでのＳＡＶＥ／ＬＯＡＤ命令のほかに、ＢＧグラフィックの保存と呼び出しができるＳＡＶＥＳ／ＬＯＡＤＳ命令が追加されている。この命令によって、これまでＢＧグラフィック作成機能を使わないとできなかった画面の保存が、ＢＡＳＩＣの命令でできるようになったんだ。プログラムとＢＧグラフィックをいっしょにＳＡＶＥすることも可能になったぞ。

```
SAVES
```

Ｖ３のＢＧグラフィックＬＯＡＤ

Ｖ３のＢＧグラフィックＳＡＶＥのところでも書いたように、ＬＯＡＤに関してもＢＧグラフィックの呼び出しができるようになった。また、ＬＯＡＤの場合は、これまでのファミリーベーシックで作ってカセットテープに保存してあったＢＧグラフィックも、ＬＯＡＤＳ命令を使って読み込むことができるんだ。もちろん、プログラムとＢＧグラフィックを同時にＬＯＡＤすることも可能だ。

```
LOADS
```

Ｖ３のバックアップＳＡＶＥ

Ｖ３ではプログラムとＢＧグラフィックの両方を同時にバックアップできるんだ。まずＢＧＧＥＴ命令でＢＧグラフィックを保存する準備をし、その後でＢＡＣＫＵＰ命令を使い、画面の指示に従ってバックアップスイッチをＯＮにする。これであとは電源を切っても大丈夫だ。また、プログラムだけ、あるいはＢＧグラフィックだけをバックアップすることもできるぞ。

```
BGGET
BACKUP
```

Ｖ３のバックアップＬＯＡＤ

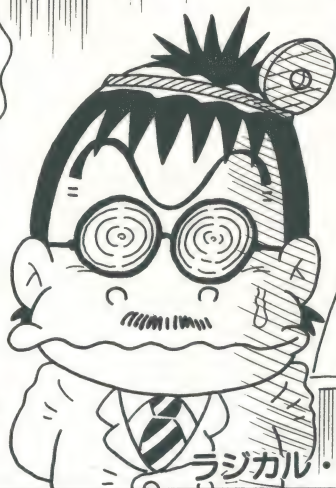
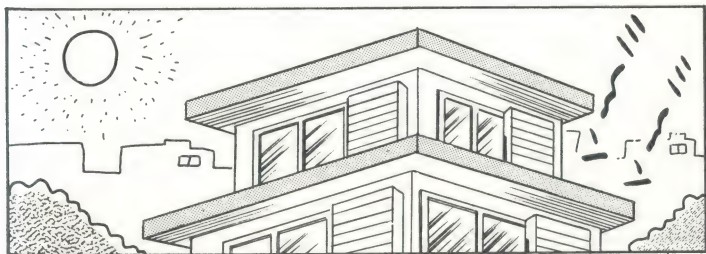
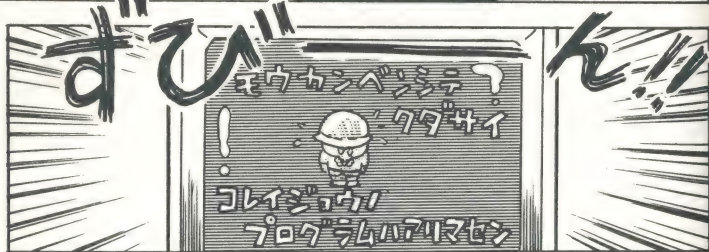
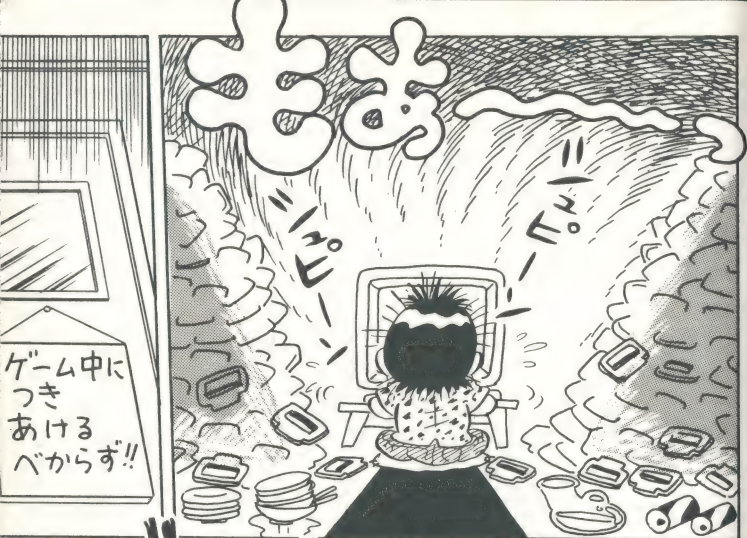
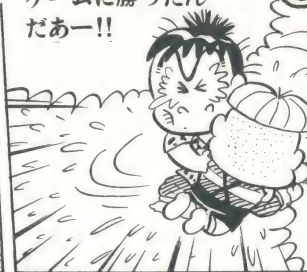
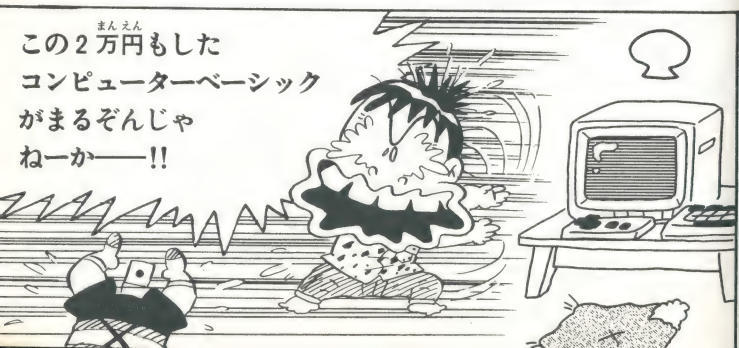
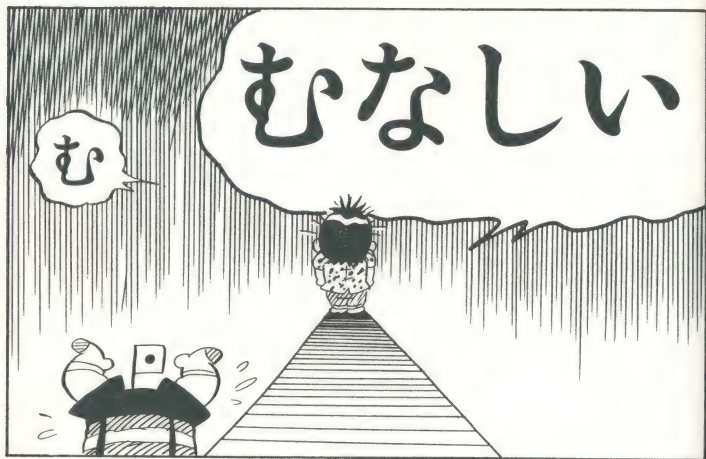
まずＶ３カセットを本体に差し込んで、電源スイッチを入れる。すると「バックアップスイッチをＯＦＦニシテクダサイ」と表示されるから、その通りにする。これでＢＡＳＩＣの状態になり、記憶されていたプログラムやＢＧグラフィックが使えるようになるぞ。なお、バックアップスイッチをＯＮにした状態で、電源スイッチを入れてスタートさせたときには、画面に「ＨＯＴ　ＳＴＡＲＴ」と表示されるぞ。

```
BGPUT
```

オリジナルゲームをつくるのだ!!

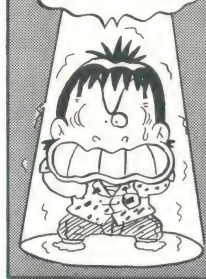
フッフッフッ

ラジカル・カンパニー

もぁ〜〜っ
ジュピーン
ジュピー
ゲーム中に
つき
あけるべからず!!
ずび〜ん!!
モウカンベンシテ!クダサイ!
コレイジョウ!
プログラムハアリマセン
フッフッフッ…
コトッ
お……
おわったか
……

コンピュータゲームに
とりつかれて10ヶ月
一歩も外に
出ずに　やり
ぬいて
おおおお
ぬおお
ついに
すべての
ゲームを
征服した
ぞ〜〜〜〜!!
ひしっ!!
やったぞカサタロー
オレはすべての
ゲームに勝ったん
だあー!!
うわーん
これで　また
いっしょに
ゲームセンター
行けますね
ガヌッ
やりましたぜ
あにき!!
おっ!?

む

むなしい

どうしたんですか
あにき~~~~~
考えてみれば
もうやる
ゲーム
が
ないって
ことは
……

この2万円もした
コンピューターベーシック
がまるぞんじゃ
ねーか──!!

ゲームソフトだって高かったんだぞ〜!!
けっきょく何にもならないじゃないか〜〜〜!!
そんなことに10ヵ月も気がつかなかったのかしら〜〜?
がが〜ん
がが〜ん
ちくしょーこうなったら……
10ヵ月かかってもおわらない超むずかしいゲームを
オレがつくってやろーじゃんか!!
オリジナルゲームなんてつくれるのかな〜〜〜?
う〜む本当にこんなアホな人に

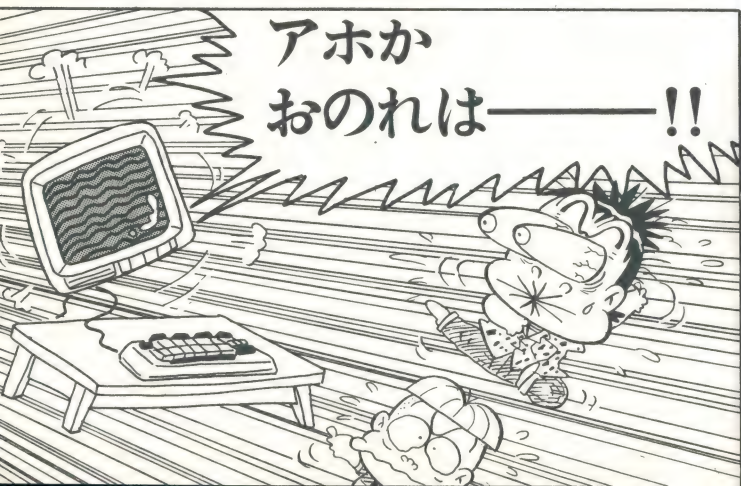
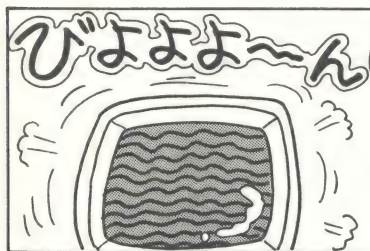
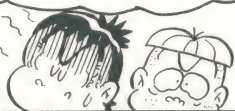
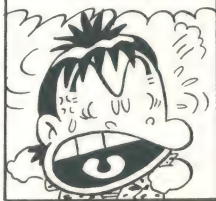
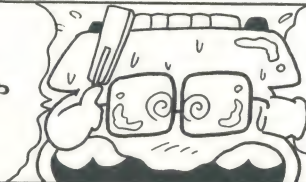
んじゃ
さっそく
やろー
かなっと

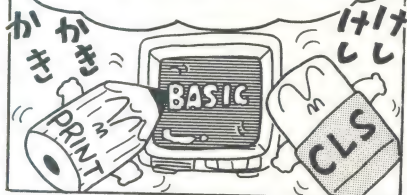
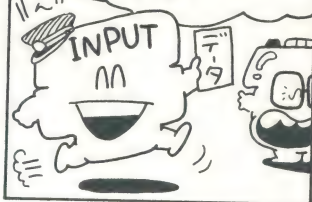
カチャカチャ
カチャカチャ

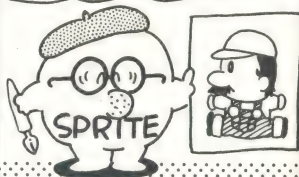
AMILY COMPUTER

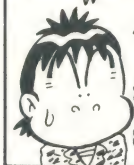
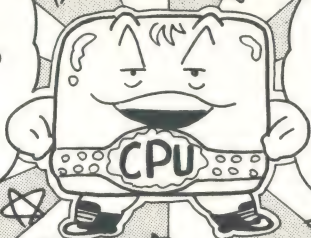
びよよよ〜ん

おっ!?　もうできたのかな――!?

アホか
おのれは―――!!

むくっ
トコトコ
な…なんでい
おまえは〜？
あ、こりゃどーも
あいさつが
おくれまして
わてベーシックの
ベーやん
いいまんねん
大橋巨泉
ではありません
しっかしあんたの
プログラミング
ありゃムチャクチャ
でござりますがな
わてが教え(おし)まひょ
けっ!!
だれがてめえ
なんかに
おそわるけえ!!

ホラ、コンピュータのサイコロゲーム!!
ゴロン
ゴロン
で…できらい……
ほ〜じゃあんたひとりでゲームつくれるんでっか?
ページもないことやしわてが教えますわ
そ…そんなんじゃあきまへん
部下!!
コマンド
わて、ベーシックはたくさんの部下から成り立ってまんのや!!
ふ〜〜ん

まずプリントくんは
画面に字を書いてくれ
クリスくんはそれを消して
くれまんねん
かきかき
けしけし
BASIC
PRINT
CLS
ランちゃんは
つくったプログラムを
うごかして
くれますねん
プログラム
RUN
ヘンスーさんは数字や
文字をきちんと覚えてくれ
ケーさんは計算をしてくれ
ますねん
変数
計算
インプットくんは
キーボードからの
データを入力して
くれます
ゆうびーん
INPUT
データ
イフさんは
いろんなことを
正確に
判定して
くれ——
あんたが正しい!!!
IF
GOTO
あらよっと
ゴトーくんは
ルーチンの
あちこちへ
いってくれますねん

プレイちゃんは
ミュージシャンで
ゲームの中のいろんな
音を出します
PLAY

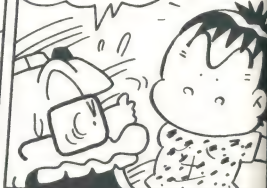
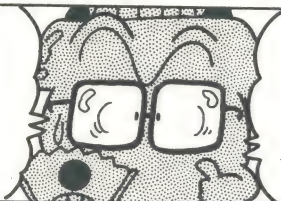
スプライトくんは
マリオなどの
キャラクタを出して
くれますねん
SPRITE

バグとエラー
こいつらは
プログラムに
まちがいが
あると
プログラムを
とめてしまう
ちゅう
えげつない
やつらや!!
エラー
バグ

ふーん
じゃあ
おまえが
いちばん
エライのか？
そや
おまへん
わては
ましんごくん
っちゅう
足のはやい
だけの
アホな人の
下で働いて
ますねん
ましんご

この人はコンピュータの中心ともいえるエライ人だす!!
そしてましんごくんの上にいるのがCPUさん!!
CPU
いやいやそんなことおまへんまず行番号っちゅう数字をかきこんで……
5 CLS
10 SPRITE
20 CGSET
21 DEF SP
+CHR
命令をつづけてかいてー
う〜んなんだか　いっぱいあってむずかしそーだな〜〜
10 SPRITE ON
20 DEF SPRI
30 INPUT "Y
40 INPUT "X
50 SPRITE 0,
60 GOTO 30
行は10、20、30、というふうに順に入力してけばべんりや!!
そして　その行の命令がおわったらRETURNキーをおせばええんや
RETURN

と、まぁこれが
だいたいの
基本や

あっそう
んじゃ　おまえ
オレのリクエストに
あわせてゲーム
つくっといてね

じゃね

ちょ…ちょっと
またん
かいな!!

わしらはただの
キカイや
人間が何も
しなければ
うごけまへん

つまり、
いっちゃん
えらいのは
あんたら
人間だっせ!!

そぉーかぁー
オレがエライ
のか〜〜
がハハハ

ぬお──し
いっちょバリバリの
ゲームつくったろ
か〜〜い!!

なんちゅー
たんじゅんな
やっちゃ

ぶいーっ!!

コンピュータに計算させよう！

ファミリーベーシックで色々なむずかしい計算をしたいと思っている人もいるだろう。

計算なら計算機を使えばいいじゃないかなんて言う人がいるだろうけど、ファミリーベーシックは計算機では、しにくい計算もやってくれたり、ほかの命令を合わせることで、計算する数値を入力してすぐ答えを出してくれる様な計算のプログラムを作ることが出来るんだ。

まずは、四則演算

四則演算とは、たし算、ひき算、かけ算、わり算の四つのことを言って、これが計算の基本になるんだ。コンピュータの場合、キミ達が使っている計算の記号が、かけるが、〝×〟から〝＊〟に、わるが、〝÷〟から〝／〟へ変わっているんだ。

```
PRINT 1+2
PRINT 2-1
PRINT 1*2
PRINT 2/1


PRINT 1+2
 3
OK
```

コンピュータに計算させよう

コンピュータに計算をさせたいのだが、答えを画面に出す方法が分らない人がいると思うんだ。

ベーシックの命令でＰＲＩＮＴ命令と言うのがあって、文字や数字を画面に出す命令を使うんだ。この命令を計算させるだけで使うのはめんどうだ。と、言う人がいたらＰＲＩＮＴのかわりに？　マークを使うといいゾ。

プログラムを作る時もＰＲＩＮＴは？　と省略出来るゾ。

計算する時の注意

ベーシックの場合、大きな数の計算が出来ないんだ。

－３２７６８から＋３２７６７までしか計算出来ないので大きな数の計算をする時には注意すべし。

あと、わり算をする時に小数点は出てこないゾ。

```
PRINT 1+2
 3
OK
? 1+2
 3
OK


PRINT 5/3
 1
OK
PRINT 10/4
 2
```

変数を使おう

ベーシックのプログラムで〝I＝3″とか〝A＝3／I″なんてしているのを見たことがあるだろう。このAとかIなんかは、変数と言ってその文字の中に数字を憶えさせることが出来るんだ。ためしに、〝A＝10″と画面に入力して[RETURN]キーを押してから〝PRINT　A″として、[RETURN]キーを押してごらん。画面に10と出ただろう。これはAと言う変数の中に10と言う数字が記憶されたことになるんだ。

```
A=10
OK
PRINT A+A+A+A
 40
```

変数で計算を

計算をする時にいくつも同じ数字を出さなくてはならない時が有るだろう。

又、同じ計算式の問題で数値だけちがうなんてことあるだろう。こんな場合に、変数がとても便利なんだ。

たとえば9＊9＊9と言う計算をA＝9としてから、A＊A＊Aにすると同じ答えになる。もしかける数が9でなくても〝A＝″の所をほかの数字にかえるだけでカンタンに計算出来る。

```
A=9
OK
PRINT A*A*A
 729
```

配列変数を使う

配列変数は同じ計算をいっぱいする時に良く使われるんだ。前の計算のプログラムの様に計算する数をかえるたびにプログラムしなおしたり、ダイレクトに変数に書き込んでいたのでは時間がかかるだろう。

そんな場合は配列変数を使ったプログラムを作るのがいいんだ。下のプログラムは、かけ算の数値を10入力して一度に画面に表示させてくれるものだ。

```
10 DIM A(10)
20 FOR I=0 TO 10
30 A(I)=I
40 PRINT A(I)
50 NEXT
```

```
10 DIM A(10)
20 FOR I=1 TO 10
30 INPUT A(I)
40 NEXT
50 FOR I=1 TO 10
60 A(I)=A(I)*5
70 NEXT
80 FOR I=1 TO 10
90 PRINT A(I)
100 NEXT
```

ＩＦ文で判定する

ＩＦ文は、ベーシックの中でもかなり重要な命令で、プログラムの流れの中心を作っていると言ってもいいんじゃないかな。主にＩＦ文はプログラムの中で数の変化があった時に反応して、プログラムの流れを変えてくれるんだ。

〝10　ＩＦ　Ａ＝10　ＴＨＥＮ　ＢＥＥＰ〟と入力してＡの変数に好きな数字を入力してＲＵＮしてみてくれ、Ａ＝10にした時にだけＢＥＥＰがなるはずだ。

```
10 A=10
20 IF A=10 THEN BEEP
```

```
10 CLS
20 INPUT "A= ";A
30 INPUT "B= ";B
40 IF A=B THEN PRINT "A=B"
50 IF A<B THEN PRINT "A<B"
60 IF A>B THEN PRINT "A>B"
70 IF A<>B THEN PRINT "A<>B"
80 IF A THEN PRINT "A<>0"
90 IF B THEN PRINT "B<>0"
100 IF A=0 THEN PRINT "A=0"
110 IF B=0 THEN PRINT "B=0"
120 GOTO 20

_
```

ＩＦ文は色々使える命令

前のページの表で色々な判定の記号の使い方が書いてあったけど、このほかにもＡＮＤ、ＯＲ、ＸＯＲなどの判定文が使えるんだ。ＩＦ　Ａ＝10　ＴＨＥＮ　ＢＥＥＰと言うプログラムは〝Ａが10ならば音を出せ〟と、言うものだけど、もし〝Ａが10でＢが20の時に一〟なんてしたい時は、ＡＮＤ命令を使い、ＩＦ　Ａ＝10　ＡＮＤ　Ｂ＝20　ＴＨＥＮ　ＢＥＥＰと、すればいいんだ。表を見て色々判定文をためしてくれ。

```
10 A=10
20 IF A=10 THEN BEEP
```

〔関係演算子〕	〔意味〕
＝	両辺が等しい
<>	両辺が等しくない（≠） ><の書きかたは使用できません。
>	左辺が右辺より大きい
<	左辺が右辺より小さい
>＝	左辺が右辺より大きいか等しい（≧） ＝>の書きかたは使用できません。
<＝	左辺が右辺より小さいか等しい（≦） ＝<の書きかたは使用できません。

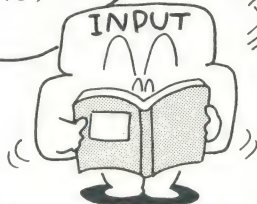
ベーシックのベーシック

基本命令

この項目で出てくる命令語

PRINT	LOCATE	INPUT
INKEY	STICK	GOTO
IF/THEN	DIM	

ふえ〜〜〜い
さーて今回は
もっとくわしく
説明して
いきまひょか!!
これも運命と
思って
あきらめるしか
あれへん…
こ…こんな
アホがわての
持ち主だなんて
なさけない…
ベーシックの
命令は
大きく4つに
わけられま〜す
入力命令
出力命令
判定命令
ジャンプ命令

LOAD
カセットから
プログラムを
読みこむ機能のこと
入力命令っちゅーのは
インプットくんや
ロードくんのように
数字や文字を
読みこんでいく
命令のことだす
INPUT
LOAD
出力命令っちゅーのは
プリントくんやプレイちゃん
画面に文字を出したり
音を出したりする
命令のことだす
SAVE
プログラムをカセット
に記録する機能
のこと
GAME
SAVE
PLAY
PRINT

判定命令っちゅーのは「もし〜ならば〜」というふうに
判決をくだす!!
何かを判定する命令のことだんねん
MOVE(0)=
T1=1 GOS
T1=0 AND
ジャンプ命令っちゅーのはプログラムの行から他の行の実行をうつす時に使う命令のことですねん
GOTO
ぴょーん!!
GOSUB プログラムの内のサブルーチンをよびだす機能のこと
およびだしを申しあげます!!
GOSUB
500 IF XPOS(0) >210
ま、ほかの命令もこの内のどれかに入りまんなー
なーーにがえらそ〜に

次はケーキと
ジュース
買って
こ〜〜い!!

オラオラ
肩もまん
か〜〜い!!

この人
ちっとも　わいの
いうこと理解
しとらんな〜

PRINT文で名前を書いてみよう

ＰＲＩＮＴ命令は、数字や文字を書く命令で、ベーシック
の中で一番、初めに覚える命令でもあるんだ。
ＰＲＩＮＴ　〝ＡＢＣ〟として[ＲＥＴＵＲＮ]キーを押してみ
てくれ、画面にＡＢＣと、出ただろう。ＡＢＣの所に自分の
名前を書くとキミの名前が画面に出てくるはずだ。
前の変数の所でも変数について覚えてもらったけど、変数
を使って画面に名前を出すプログラムが下にある。

```
10 A$="KATAYAMA YASUHIKO"
20 PRINT A$
```

LOCATE

文字や数字を自由にプリント出来る様になってくると、画
面にレイアウトしながら出力したくなるだろう。そんな時は
ＬＯＣＡＴＥ命令を使うんだ。ＬＯＣＡＴＥ命令は画面の座
表(たて０～23、よこ０～27)の好きな場所に数字や文字をプ
リントしたい時、この命令の後にＰＲＩＮＴ命令などを入力
すると、その場所に出してくれるゾ。下のプログラムのＸ
(よこ)、Ｙ(たて)の変数を色々かえてＲＵＮしてみよう。

```
20 X=3
30 Y=10
40 LOCATE X,Y
50 A$="MARIO"
60 PRINT A$
```

ＩＮＰＵＴで文字を読み込め

文字や数字を変数に憶えさせる時にいちいちプログラムをなおしていたんでは、時間も手間もかかる。そんな時にＩＮＰＵＴ命令を使うと、とても便利だ。ＩＮＰＵＴは、キーボードから押された数字や文字を変数の中に憶えさせてくれる命令で〝アナタノ　ナマエハ？〞なんて、コメントをプリントしてくれることも出来るんだ。下のプログラムは、キーボードからキミの名前を入力して画面に出すプログラムだ。

```
10 INPUT "アナタノ ナマエハ ?" , A$
20 PRINT "アナタハ " ;
30 PRINT A$ ;
40 PRINT " サン "
```

ＩＮＫＥＹでキーの読み込み

リアルタイムのゲームなどで、キーを押して反応する様なプログラムを作る時に、リアルタイムにキーボードの押されたキーを読み込んで、変数に憶えさせてくれるのがＩＮＫＥＹ命令なんだ。下のプログラムは、〝Ｅ〞のキーが押されるとＢＥＥＰがなってプログラムが終わる様になっているけど、ＩＦの〝Ｅ〞と判定する所を色々な文字に変えてみよう。

ほかにもジョイスティックの読み取り命令がある。

```
10 I$=INKEY$
20 IF I$ <> "E" THEN 10
30 BEEP
40 END
```

ＩＦ命令でスティックの方向を

ゲームを作る時にどうしても、キーよりスティックを使ったゲームを作りたくなるだろう。ＳＴＩＣＫ命令はジョイスティックの方向を変数に憶えさせてくれる命令なんだ。(注、ＩＮＫＥＹでは文字変数〝$〟を使ったけどＳＴＩＣＫは数字として方向をかえしてくるので、文字変数は使えません)

下のプログラムはスティックＩからキーの押した方向をＩＦ文で判定してプリントしてくれるプログラムだ。

```
10 IF STICK(0)<>0 THEN PRINT
 STICK(0)
20 GOTO 10
```

```
LIST
10 I=STICK(0)
20 IF I=0 THEN 10
30 IF I=1 THEN PRINT "1:ミギ"
40 IF I=2 THEN PRINT "2:ヒダリ"

50 IF I=4 THEN PRINT "4:シタ"
60 IF I=8 THEN PRINT "8:ウエ"
OK
RUN
1:ミギ
OK
RUN
2:ヒダリ
OK
RUN
4:シタ
OK
RUN
8:ウエ
OK
█
```

GOTO文でくりかえし処理

同じプログラムをくりかえして実行する時に、いちいちRUNしていたんでは、めんどうだネ。そんな場合、GOTO命令と言って、指定した行に処理をうつしてくれる命令を使うんだ。

下のプログラムは、AとBをたし算してプリントするプログラムだけど、最後の行でGOTO 10として、くりかえしのプログラムにしている。

```
:
:
:
50 GOTO M
```

```
LIST
10 INPUT "A=";A
20 INPUT "B=";B
30 C=A+B
40 PRINT "A+B=" ; C
50 GOTO 10
OK.

RUN
A=25
B=7
A+B= 32
A=
```

GOSUBでサブルーチンに

大きなプログラムで同じルーチンをたくさん使う場合に、その同じルーチンをいくつもプログラムしていたんでは、プログラムが大きくなりすぎたり、メモリが足らなくなったりするだろう。そんな使用度が高いルーチンを一つにまとめて、プログラムの中の、どこからでも実行してその行にもどしてくれるのがＧＯＳＵＢ命令と、ＲＥＴＵＲＮ命令なんだ。

注：サブルーチンの最後には必ずＲＥＴＵＲＮとつけること。

```
10 GOSUB M
```

```
10 CLS
20 GOSUB 50
30 GOSUB 80
40 GOTO 20
50 X=RND(27)
60 Y=RND(22)
70 RETURN
80 LOCATE X,Y:PRINT CHR$(RND
(3)+253)
90 COLOR X,Y,RND(3)
100 RETURN
```

FORでくりかえしプログラム

GOTO文でくりかえして実行してやるプログラムを作ったけどこれではいつまでたっても、プログラムは終らないよネェ。それに10回だけくりかえせばいいんだなんて、言う場合も出てくると思う。そんな時は、下のプログラムの様に、くりかえす行の初めにFOR命令を使い、初めのFOR命令にもどるために、最後にNEXT命令を付けるのだ。この2つは必ずいっしょにつける。

```
10 FOR I=M TO N
 :
 :
 :
60 NEXT
```

```
LIST
10 FOR I=1 TO 10
20 INPUT "A=";A
30 INPUT "B=";B
40 C=A*B
50 PRINT "A*B=" ; C
60 NEXT
OK.
RUN
A=34
B=6
A*B= 204
A=90
B=45
A*B= 4050
A=
```

RNDでかずあてゲーム

ゲームでキャラクタがメチャクチャに動いたり、ルーレットゲームなどで予想が出来ない数字を出したりするのがRND命令なんだ。このRND命令は、指定された数値によって乱数を発生させるんだ（乱数と言うのは、メチャクチャな数のことなんだ）。この命令を使って1～9までの数あてゲームを作ってみた。当たるとBEEPがなるゾ。

又、RNDの9の数字を色々かえてみよう。

```
10 I=RND(M)
  :
  :
```

```
LIST
10 I=RND(9)
20 PRINT "*** カズアテ ゲーム ***"
30 INPUT J
40 IF I<J THEN PRINT "オオキイ":
GOTO 30
50 IF J<I THEN PRINT "チイサイ":
GOTO 30
100 PRINT "アタリ"
110 BEEP
120 END
OK
RUN
*** カズアテ ゲーム ***
?5
オオキイ
?3
オオキイ
?1
チイサイ
?2
アタリ
OK
```

ＤＩＭで配列を

計算の所で変数の配列を使ったプログラムの中にＤＩＭ～と言う行があったけど、この行は、コンピュータに〝これから変数の配列をとりますヨ〟って言う命令で、配列を使ったプログラムの時にこの命令を実行しておかないと、〝配列をとっていない〟と、エラーが出てしまうゾ。下のプログラムは、10個数字を入力し終わったらその数字を全部出してくれるプログラムだ。

```
10 DIM A(M)
 :
 :
```

```
LIST
10 DIM A(10)
20 FOR I=1 TO 10
30 PRINT I ; " バンメ " ;
40 INPUT A(I)
50 NEXT
60 PRINT
70 FOR I=1 TO 10
80 PRINT I ; " バンメ ハ " ; A(I
)
90 NEXT
OK
RUN
 1 バンメ ?1
 2 バンメ ?3
 3 バンメ ?5
 4 バンメ ?7
 5 バンメ ?
```

DATA文でデータの読み込み

プログラムでいちいち数字を憶えさせるのはめんどうで、プログラムの中にあらかじめ、数字を入れておきたいなんて思っている人もいると思う。そんな時のためにDATAと言って、プログラムの中にあらかじめ数を入力させておく方法があるんだ。こうすればいちいちキーからインプットしなくても良くなるゾ。READ命令はDATAから変数の中に憶えさせてくれるDATA文にはなくてはならない命令だ。

```
 :
50 DATA M
60 DATA N
 :
```

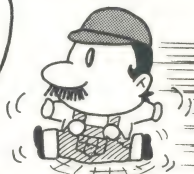
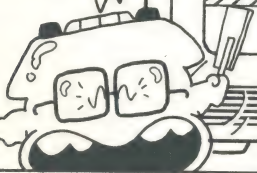
```
10 FOR I=0 TO 4
20 READ A$
30 PRINT A$
40 NEXT
50 DATA A
60 DATA ABC
70 DATA ABCD
80 DATA ABCDE
90 DATA ABCDEF
```

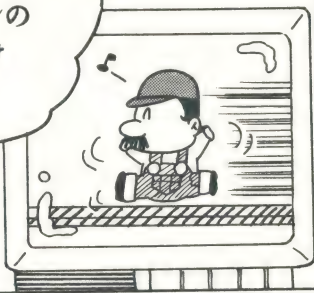
ベーシックのベーシック

MOVE（ムーブ）命令（めいれい）

この項目（こうもく）で出（で）てくる命令語（めいれいご）

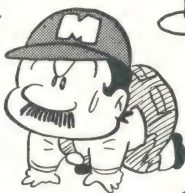
DEF MOVE	XPOS	VCT
MOVE	YPOS	CRASH
SPRITEON	CUT	PALET
ERA		

お――
だんだん
おもしろく
なって
きたぜい!!
MOVEというのは
マリオなどの
キャラクタを動かす
ことでゲームが
おもしろくなるも
ならないも
ここにあります!!
それにはまず
DEFやSPRITEで
キャラクタの動く方向や
スピード・距離などを
指定すると
DEF
アニメキャラクタ
の動きをきめる
機能のこと
おまえはネ
ああいって
こういって
....
SPRITE
まりお
DEF

このよーに
アニメーションの
ように動くわけ
ですねん

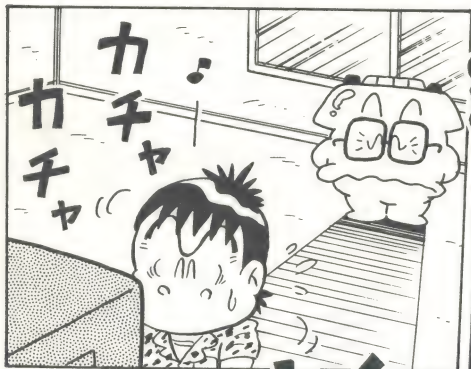
わーーっ
手品みたい
〰〰〰

このほかにも
キャラクタの動きを
とめるCUT命令や

まった!!

CUT

いーかいきみは
あーいって
こーいくんだよ

POSITION

動かす前に　動きはじめの方向を
指定するPOSITION命令などが
ありますねん

ちゅーわけで　あんたらも
このMOVEをやって
みなはれ!!

おーっ!!

えっ!?
よーし!!
これで　動くぞ
──!!
ホラ
ゴロゴロ
わいが
あまかった
どこまで
アホなんや
この人は〜〜!!
いったい
ボクは
何のために
出てるんだろう
……

スプライトでマリオを出そう！

まず、SPRITE　ON RETURN を入力しよう。これはマリオを画面に登場させる準備なんだ。画面にOKと表示が出たら、次にDEF　SPRITE命令をおこなう。これは、４つのキャラクタで構成されているアニメキャラクタを指定する命令なんだ。リスト通り、まちがいなく入力しよう。

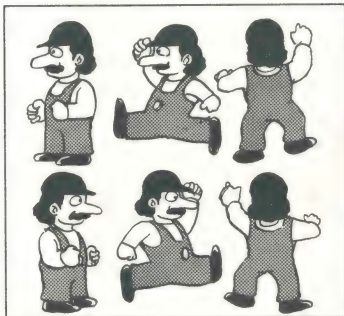
そしてSPRITEと入力し、その後にスプライト番号、ヨコの座標、タテの座標を入力する。これで、指定した座標の位置にマリオが姿を現わすぞ。

```
LIST
10 SPRITE ON
20 DEF SPRITE 0,(1,1,1,0,0)=
CHR$(0)+CHR$(1)+CHR$(2)+CHR$
(3)
30 SPRITE 0,100,30
OK
RUN
OK
```

こんなマリオも出せるんだ

スプライトでマリオを出したら、こんどは下のリストを入力してみよう。これでRUNを入力して RETURN キーを押すと、画面にマリオが6つ表示されるはずだ。でも、よく見てみると、このマリオはどれも違ったポーズをしているぞ。左を向いていたり、右を向いていたり、後ろを向いていたり……。

そう、マリオのポーズは1つじゃなくて、いろいろあるということなんだ。

このポーズによって、マリオはゲームのとき方向を変えたりするんだよ。

```
10 SPRITE ON
20 FOR I=0 TO 5
30 K=I*4
40 DEF SPRITE I,(2,1,1,0,0)=
CHR$(K)+CHR$(K+1)+CHR$(K+2)+
CHR$(K+3)
50 NEXT
100 FOR I=0 TO 5
110 K=I*20+50
120 L=I*10+130
130 SPRITE I,K,L
140 NEXT
OK
RUN
OK
```

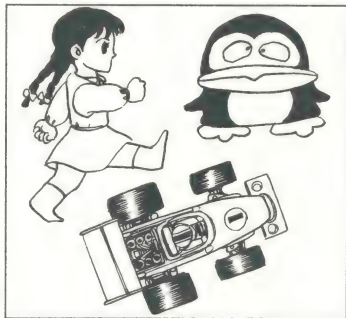
マリオに好きな色をつけよう！

マリオが画面に登場したら、CGSET命令を使ってみよう。マリオの色が、最初に登場したときの色から変化するぞ。さらに、PALETS命令を入力してみよう。すると、少しちがった色のマリオが表示されるぞ。次にLISTを入力。これで、さっき入力したPALETS命令が記憶されているかを確認するんだ。行番号40のPALETSを入力すると、前のCGSETのプログラムが書きかえられるぞ。PALETS命令の数値を変えればマリオの色も変わるんだ！

```
LIST
10 SPRITE ON
20 DEF SPRITE 0,(2,1,1,0,0)=
CHR$(0)+CHR$(1)+CHR$(2)+CHR$
(3)
30 SPRITE 0,30,170
40 PALETS 2,13,32,12,4
OK
RUN
OK
■
```

こんなキャラクタも出せるぞ！

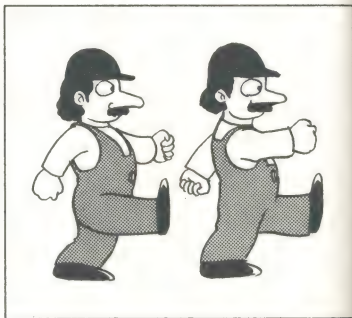
ファミリーベーシックで、登場させることのできるアニメキャラクタは、マリオだけじゃないぞ。ＤＥＦ ＭＯＶＥ（ｎ）＝ＳＰＲＩＴＥ（Ａ、Ｂ、Ｃ、Ｄ、Ｅ、Ｆ）を入力するとき、Ａのところでアニメキャラクタの種類を選ぶことができるんだ。

用意されたキャラクタは全部で16種類。マリオのほかに、レディ、ファイターフライ、アキレス、ペンペン、ファイアーボール、車、スピナー、スターキラー、スターシップ、爆発、ニタニタなどがあるぞ。

```
10 SPRITE ON
20 CGSET 1,2
30 FOR I=0 TO 7
40 DEF MOVE(I)=SPRITE(I,I,1,
20,0,0)
50 MOVE(I)
60 NEXT
```

マリオを動かすぞ！

マリオを動かすには、MOVEコマンドを使おう。まず、DEF　MOVEでアニメキャラクタの種類と動きを定義し、その後、MOVE命令を入力して、動きを開始させるんだ。

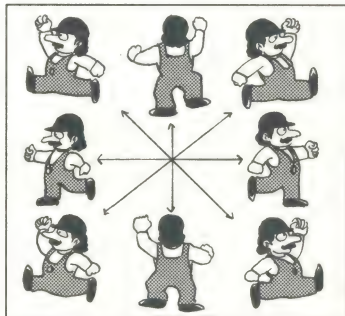
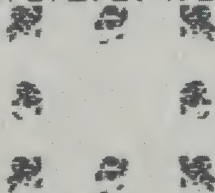
キャラクタを動かせる方向は全部で８つ。真上が１、右上が２、右が３、右下が４、下が５、左下が６、左が７、左上が８、というように数値でその方向を指定するんだ。

下のリストは、マリオを左から右へ移動させるためのもの。さっそくキミもためしてみよう。

```
LIST
10 SPRITE ON
20 DEF MOVE(0)=SPRITE(0,3,4,
80,0)
30 POSITION 0,30,150
40 MOVE 0
OK
RUN
OK
```

8方向にマリオを動かそう！

マリオは1画面に最高8人まで登場させることができる。しかも同時にそれぞれを別方向に動かすことができるんだ。

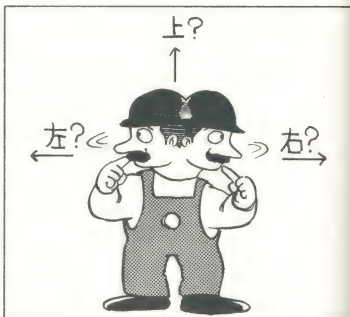
ここでは、下のリストのように、MOVE命令を使ってマリオを8方向に動かしてみよう。

まず、NEW RETURNを入力し、前のプログラムをメモリから消しておく。そしてリスト通りにプログラムを入力。最後にRUN RETURNを入力しよう。これで、8人のマリオがそれぞれ8つの方向に動き始めるぞ。

```
LIST
10 FOR I=0 TO 7
20 DEF MOVE(I)=SPRITE(0,I+1,
4,240,0)
30 POSITION I,120,120
40 NEXT
50 SPRITE ON
60 MOVE 0,1,2,3,4,5,6,7
OK
RUN
OK
```

ＶＣＴで動いている方向を

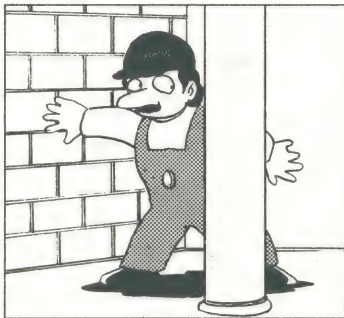
スプライトのＭＯＶＥ命令を使っているとどうしても、キャラクタの動いている方向を知りたくなるよネ。

Ｖ３のベーシックは、そんな人のためにＶＣＴと言う命令が加えられたんだ。

このＶＣＴは、キャラクタの番号を指定してあげることによって、そのキャラクタが、今どこの方向を向いているのかをおしえてくれる関数なんだ。

下のプログラムは、そのテストプログラムだ。

```
10 SPRITE ON
20 CGSET 1,2
30 INPUT "ウゴカス ホウコウ ",M
40 DEFMOVE(0)=SPRITE(0,M,3,3
0,0,0)
50 MOVE 0
60 PRINT "ウゴイテイル ホウコウ ",VCT(
0)
70 IF MOVE(0)=-1 THEN 70
80 GOTO 30
```

かくれんぼマリオ

ファミリーベーシックの画面というのは、前からスプライト面(前)、バックグラウンド面、スプライト面(後)、バックドロップ面というふうに構成されている。ここで面白いのは、スプライト面がバックグラウンド面の前後に2つあることだよね。つまりスプライト面っていうのはバックグラウンド面の前にも後ろにも設定できるんだ。その設定はＤＥＦ　ＭＯＶＥ命令でおこなう。表示の優先度を1にして後ろのスプライト面を使えば、マリオの姿を背景の後ろにかくせるぞ。

```
LIST
5 CLS
10 FOR I=0 TO 7
20 DEF MOVE(I)=SPRITE(0,I+1,
4,240,1)
30 POSITION I,120,120
40 NEXT
50 SPRITE ON
60 FOR I=1 TO 40
70 LOCATE RND(27),RND(23)
80 PRINT "█████"
90 NEXT
100 MOVE 0,1,2,3,4,5,6,7
OK
```

動いているマリオを消そう！

ゲームをよりいっそう面白くするためには、アニメキャラクタを画面から消すことも大切なテクニックだよね。そのためにはＥＲＡ命令を使おう。この命令を使うときは、ＥＲＡと入力した後に、消したいマリオの番号を入力しておく。

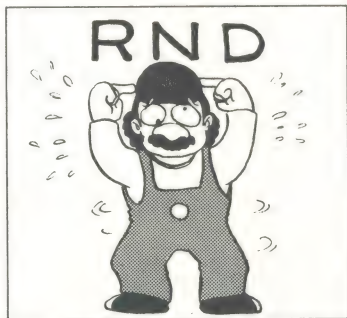
そして、ＲＵＮでマリオを歩き出させ、ＳＰＥＣＥキーを押す。すると、指定した番号のマリオが画面から姿を消すぞ。

消えたマリオを再び画面に出したいときは、ＭＯＶＥ命令を使う。指定した番号のマリオが再登場だ。

```
10 SPRITE ON
20 CGSET 1,2
30 CLS
40 INPUT "ケシタイ マリオ ",C
50 FOR I=0 TO 7
60 DEFMOVE(I)=SPRITE(0,3,3,2
55,0,0)
70 POSITION I,0,30+I*20
80 MOVE I
90 NEXT
100 IF INKEY$<>" " THEN 100
110 ERA C
120 IF INKEY$<>"E" THEN 120
130 MOVE C
```

色々な方向に動くマリオ

ベーシックのゲームでよくペンペンやニタニタが色々な方向に動いて、プレイヤーのじゃまをするなんて言う ゲームがあるだろう？　こう言うプログラムには、かならずと言っていいほど、キャラクタの動きにＲＮＤ命令が使われているんだ。

下のプログラムは、ＤＥＦ ＭＯＶＥのＳＰＲＩＴＥの設定の時に動く方向をランダムにかえてみただけなんだよ。
プログラムをＲＵＮしてみると、色々な方向にマリオが動くゾ。

```
LIST
10 CLS : CGSET 1,1
20 DEF MOVE(0)=SPRITE(4,RND(
7)+1,2,15,1)
30 POSITION 0,120,120
40 SPRITE ON
50 DEF MOVE(0)=SPRITE(4,RND(
7)+1,4,10,1)
60 MOVE 0
70 IF MOVE(0)=0 THEN 50
80 GOTO 70
OK.
```

おどろくマリオを

Ｖ２では、スプライト同士のぶつかりを判定する時、キャラクタの座標で判定させていた場合が多かったけど、Ｖ３ではＣＲＡＳＨと言う命令がついて、スプライト同士のぶつかりもカンタンに判定出来る様になったんだ。

下のリストは、右から左へ動くマリオと、左から右へ動くファイヤーボールが、画面の中心部でぶつかって、マリオがやけどをしてしまうと言うプログラムなんだよ。Ｖ３を持っているキミＣＲＡＳＨを色々使ってみよう。

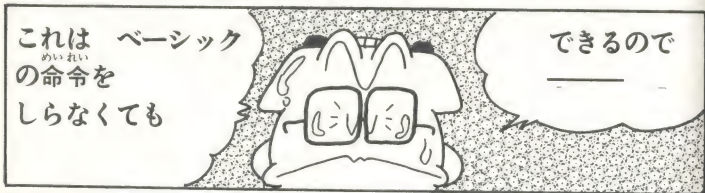
```
10 SPRITE ON
20 CGSET 1,2
30 DEFMOVE(0)=SPRITE(0,7,3,2
55,0,0)
40 DEFMOVE(1)=SPRITE(5,3,3,2
55,0,0)
50 POSITION 0,250,100
60 POSITION 1,0,100
70 MOVE 0,1
80 IF CRASH(0)=-1 THEN 80
90 X=XPOS(0):Y=YPOS(0)
100 DEFMOVE(0)=SPRITE(0,8,1,
20,0,0)
110 POSITION 0,X,Y
120 MOVE 0
```

ベーシックのベーシック

BGグラフィック

この項目で出てくる命令語

CLS
VIEW
READ
DATA
LOCATE
RED
PALET
SERECT
COPY
MOVE
CLEAR
FILE
CHAR
MODE
CGSET

BGグラフィック
ちゅーのはゲームに
でてくる　カラフルな
背景を描くことなので
あ〜〜〜る!!

ひく
ひく

ひとの説明を
まったく　きかない
キミ!!

頭の中が
カラッポな
キミ!!

とまぁ、そんな人でも
ゲームを作る気分だけ
でも味わえるっちゅう
わけだ!!

フン!!

キャラクタを使った楽しいアニメーションなんかもベーシックでできんやで
配色パターンも4色パターンあるからゲームのほかにも
ホンマにちゃんとできるんだっか？
じーっ
う～んこんどこそオレにもできそーだな!!
ひとこと多い!!
つけものいし
マンガばっかし見てっかんね!!
なにをかくそうオレは絵の才能だけはあるのだ
Glinaga

でも色なんか何色かってきまってんじゃないの？

？？？？

うへ～んええしつもんじゃ

そーゆー人にはパレット命令をおすすめします!!

この命令はひとつのキャラクタに4色から最大52色まで1配色単位で色をつけることができるんやって!!

カラーじゃないのがザンネン!!

まあキレイ!!

う〜ん　この人(ひと)が
まじめにやってると
なぜか不安(ふあん)に
なっちゃう　ボク
カミャ
カミャ
ぱっ
ぱっ!!
ぱっ!!
うわ――っ
なんだ
この
ブキミな
画面(がめん)は
――!!

げ…

げ…

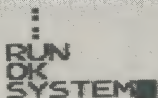
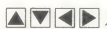
わたくし実家へ帰らせていただきます……

つまり…あの…情熱的な色がですね…

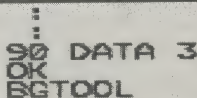
芸術はバクハツだ〜〜〜〜〜〜!!

岡本タロー

BGで背景を描くには？

まず、BGグラフィック画面を出してみよう。ベーシックを使っている場合は、SYSTEMと入力し、RETURNキーを押す。それ以外の場合は、メニュー画面でGAMEベーシックを選べばいいんだ。すると画面がゲームベーシックモード画面に変わるから、2のキーを押す。これでBGグラフィック画面が出たぞ。背景を描くには、▲▼◀▶カーソルキーを使って、キャラクタを好きな位置に置いていくんだ。

V3のBGグラフィック

いままでのファミリーベーシックでは、ベーシックのプログラムを作る機能とBGグラフィックを作る機能をメニュー画面で選ぶようになっていた。ところがV3では、BGグラフィックを作成する機能に〝BG　TOOL〟という名をつけ、ベーシックから直接呼び出せるようになっているんだ。それに、BGグラフィックが2面使えるようになったのも特徴。V3ではBGグラフィックをより幅広く使うことができるぞ。

```
:
90 DATA 3
OK
BGTOOL
```

これがBGグラフィック画面だ

ＢＧグラフィック画面の左上には四角いワクが表示されている。これがＢＧグラフィックのカーソルで、▲▼◀▶カーソルキーを使って移動させる。カーソルの動ける範囲はヨコが０から27までのカーソル28個分、タテが０から20までのカーソル21個分だ。カーソルの座標は左下のＸ(ヨコ)とＹ（タテ）に表示されるぞ。その横に表示されているのがキャラクタグループ。好きなキャラクタをこの中から選ぶんだ。

```
X:14 MODE 0
Y:20 SELECT
```

ファンクションメニュー

ESCキーを押すと、画面左上にファンクションメニューが表示されるぞ。ＳＥＬＥＣＴはキャラクタを選ぶときに使用。Dキーを押すとカーソル内のキャラクタが消える。ＣＯＰＹは、キャラクタをもう１つ別の位置にコピーするときに使う。ＭＯＶＥはキャラクタを移動するときに使用。ＣＬＥＡＲを使うとすべてのキャラクタが消える。ＦＩＬＥは、絵をＳＡＶＥ、ＬＯＡＤ時に使用。ＣＨＡＲは文字等の表示。

```
>SELECT
 COPY
 MOVE
 CLEAR
 FILE
 CHAR
```

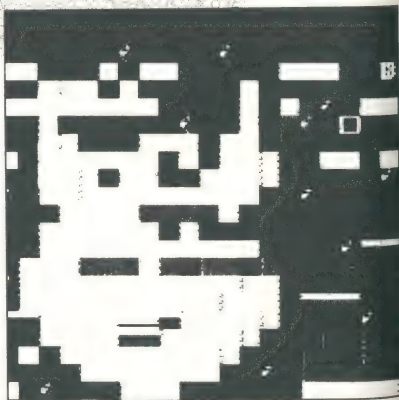
グラフィックチャートの見方

BGグラフィックでは、ヨコ28マス×タテ21行の範囲内で背景の絵が描ける。グラフィックチャートは、BGグラフィックで描く各キャラクタのデザインデータを、その範囲内に示したものだ。ここで見方をおぼえておこう。

たとえば、K50とあれば、Kグループの5のキャラクターを使い、配色は0を意味する。0の配色というのは、SELECTモードで選択する0番の配色ということだ。

```
X:00 MODE 0
Y:00 SELECT
```

BGグラフィックのキャラクタ

キャラクタは8個×13グループが用意されている。画面に表示されるのはそのうちの1グループ。CLR HOMEキーを押すとグループは順方向に変わり、SHIFTキーを押しながらCLR HOMEキーを押すと逆方向に変わるぞ。

グラフィックをカセットに

キミの作ったBGグラフィックを保存しておくために、カセットに記憶させよう。まず、ESCキーを押してファンクションメニューを表示する。その中からFILEを選び、[ ]キーを押す。すると、〝SAVE(S), LOAD(L)?〟と表示されるから、Sキーを押そう。この後に、ファイル名を記入して、テープレコーダでカセットテープにSAVEするんだ。

```
SAVE(S),LOAD(L)?
```

V3での保存方法

V3ではBGグラフィックもメモリバックアップできるんだ。NEWRETURNBGGETRETURNBACKUPRETURNとキーを押して、バックアップスイッチをONにすればバックアップの完了だ。また、カセットへの保存も、SAVES／LOADS命令が加わったのでいっそう便利になったぞ。おまけに、プログラムとBGグラフィックの本を同時にSAVE／LOADすることができるんだ。

```
BGGET
BACKUP
```

グラフィックに色を着けよう！

ＢＧグラフィック画面の下に表示されている８つのキャラクタには白と青系の色しか着いていないよね。でも、ここでＲＥＴＵＲＮキーを押してみよう。キャラクタの色が変わりＭＯＤＥの横の数字が０から１になったはずだ。さらにＲＥＴＵＲＮキーを押していくとキャラクタの色は次々と変わり、数字が３になった後、再びもとに戻る。キミの好みの色をキャラクタに着けてやろう。

```
V2 - SYSTEM
V3 - BGTOOL
```

```
X:00 MODE 3
Y:00 SELECT

X:00 MODE 2
Y:00 SELECT

X:00 MODE 1
Y:00 SELECT

X:00 MODE 0
Y:00 SELECT
```

こんなグラフィックも作れるよ

BGグラフィックを使っていると、どうしても思いどおりにグラフィックを作ることが出来ないだろう。

キャラクタがツレなかったりすると、どうしても出来ないグラフィックも出てくると思うが、パレットをベーシックで使ってグラフィックの色をかえてやったりすると意外に面白いキャラクタが、出来たりするゾ。

下のグラフィックリストを入力して見よう。

```
100 PALETS 0,C(0),C(1),C(2),
C(3)
```

```
10 SPRITE ON:CLS:DIM C(3)
20 INPUT "ｷｬﾗｸﾀｰ NO?",C
30 FOR I=0 TO 7
40 DEF MOVE(I)=SPRITE(C,I,1,
20,0,0):MOVE I
50 NEXT
60 FOR I=0 TO 3
70 IF STICK(0)=8 THEN C(I)=C
(I)-1 IF C(I)<0 THEN C(I)=61

80 IF STICK(0)=4 THEN C(I)=C
(I)+1 IF C(I)>60 THEN C(I)=0

90 IF STICK(0)=1 THEN 130
100 PALETS 0,C(0),C(1),C(2),
C(3)
110 LOCATE 0,1:PRINT C(0),C(
1),C(2),C(3)," "
120 GOTO 70
130 FOR D=0 TO 1000:NEXT
140 NEXT:GOTO 60
```

背景とアニメキャラクタを合成

背景の絵を描きおわったら、その上にアニメキャラクタを合成してみよう。まず、ESCキーを押してからSTOPキーを押し、ゲームベーシックモードに戻す。次に1キーを押し、ベーシック画面にする。そして、下図のベーシックプログラムを入力。あとは、RUN RETURNで合成が完了だ。

```
VIEW
```

```
LIST
5 VIEW
10 CGSET 1,1
20 DEF MOVE(0)=SPRITE(4,RND(
7)+1,2,15,0)
30 POSITION 0,120,120
40 SPRITE ON
50 DEF MOVE(0)=SPRITE(4,RND(
7)+1,4,10,0)
60 MOVE 0
70 IF MOVE(0)=0 THEN 50
80 GOTO 70
OK.
■
```

ベーシックからグラフィックを！

BGグラフィックは、BG画面にしないと出来ないと、思っている人がいると思うけど、ベーシックからでもグラフィックをかくことが出来るんだ。

下のプログラムはDATA文でBGグラフイックの元になるデーターが入っていて、ここをかえることによって色々キャラクタをかえることが出来るゾ。

```
  :
  :
20 CGSET M,N
```

```
LIST
5 CLS
6 CGSET 0:1
10 DATA 215,207,197,0
20 READ A
30 IF A=0 THEN END
40 FOR I=1 TO 28
50 PRINT CHR$(A);
60 NEXT
70 GOTO 20
OK
■
```

BGグラフィックをスクロール

スクロールと聞いて、まずピンとこない人もいると思うけど、ゼビウスなどで下の画面が流れていく様なことをスクロールと言うんだ。キミは、このベーシックでスクロールが出来ることを知っているだろうか？　この本でもスクロールを使ったプログラムがのっている。下のプログラムを見てくれれば分るけど、たいしたことはしていないんだ。LOCATEのY座標を23にしているのがポイントなんだ。

```
10 LOCATE RND(M),23
 :
 :
```

```
LIST
10 LOCATE RND(28),23
20 PRINT CHR$(207)
30 GOTO 10
OK.
```

COLOR命令を使おう

ベーシックでグラフィックを作ったのはいいけど、「カラーを使う時はどうするんだ」と思う人もいるんじゃないかな。そんな時はCOLOR命令を使えばいいんだけど、このCOLOR命令は全体の色を変化させたりすることが出来ない命令で、XとYの座標を指定して、BGグラフィックで使ったカラーMODEの色指定をしなければならないのだ。

```
10 INPUT "  X  --";X
20 INPUT "  Y  --";Y
 :
50 COLOR M,N,O
```

```
10 INPUT "  X  --";X
20 INPUT "  Y  --";Y
30 INPUT "COLOR -";C
40 FOR I=0 TO 700:PRINT CHR$
(255);:NEXT
50 COLOR X,Y,C
60 GOTO 10
```

PALET命令で色を変える

PALET命令を使うと、背景の色やアニメキャラクタの色を、52色の色コードの中から好きな色を選んで着色することができるぞ。PALET Bでは背景を、PALET Sではアニメキャラクタを着色できるんだ。

背景を着色するか、キャラクタを着色するかのどちらかを指定したら、次は配色番号の値を決めよう。さらに、C1～C4までの4色のコードを設定するんだ。

```
PALET
```

```
10 SPRITE ON
20 DEF SPRITE 0,(0,1,0,0,0)=
CHR$(0)+CHR$(1)+CHR$(2)+CHR$
(3)
30 SPRITE 0,100,100
40 INPUT "COLOR 1 ";C1
50 INPUT "COLOR 2 ";C2
60 INPUT "COLOR 3 ";C3
70 INPUT "COLOR 4 ";C4
80 PALETS 0,C1,C2,C3,C4
90 GOTO 30
```

カラーバーを出そう

ファミリーベーシックでは52色の色が使えるけど、これらの色をすべて見てみたいと思う人はカラーバーを出そう。そのためにはまず、下のリスト通りのプログラムを入力。このプログラムを実行させると、画面に12本のカラーバーが現われるぞ。さらに、キーボードのキーをどれか1つ押せば、カラーバーの色が変わっていき、全色が表示されるんだ。

52 COLOR CHART

```
LIST
10  A$=CHR$(254)+"  "
20  FOR I=1 TO 3
30  A$=A$+A$
40  NEXT
90  CGSET 0,0
100 CLS
110 FOR I=1 TO 22
130 PRINT A$
140 NEXT
150 FOR I=0 TO 60
160 PALETB 0,15,58,I,I
170 IF INKEY$="" THEN 170
180 FOR J=0 TO 499
190 NEXT
200 NEXT
OK.
■
```

ランダムに星空を出してみよう

ベーシックゲームを作る時、ちょっとした星空をBGグラフィックで作るのはとてもめんどうな作業だ。

出来れば、ベーシックのプログラムで星空を作りたいネ。

それに、BGグラフィック画面で作ると、どうしても星空らしく見えないだろう。

下のプログラムは、ランダムでキャラクタ座標のX、Yを出して、その場所に星のキャラクタを出すものだ。

```
RND(M)
```

```
LIST
10 CLS
20 FOR I=1 TO 30
30 LOCATE RND(27),RND(22)
40 PRINT CHR$(205);
50 LOCATE RND(27),RND(22)
60 PRINT CHR$(206);
70 NEXT
80 PALETB 0,15,48,RND(59),RN
D(59)
90 FOR I=1 TO 500:NEXT
100 GOTO 80
OK.
```

BGグラフィックでアニメーション

BGグラフィックを作ったりしていると、自分のキャラクタが動いたりしたらいいな、なんて思う時があるだろう。

そんな時は、パレットを使うといい。

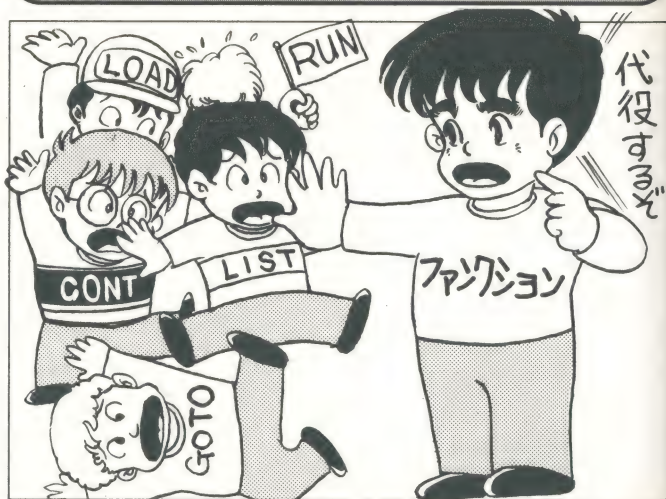
パレットは、キャラクタの色を好きな様に変えられる命令だけど、キャラクタの色の指定を黒にしたりしなかったりすることでアニメーションを作ることが出来るんだ。

```
PALETB M,N,O,P
```

```
10 FOR I=0 TO 643
20 PRINT CHR$(253+RND(3));
30 NEXT
40 R=RND(3)
50 IF R=0 THEN PALETB 0,13,R
ND(60),13,13
60 IF R=1 THEN PALETB 0,13,1
3,RND(60),13
70 IF R=2 THEN PALETB 0,13,1
3,13,RND(60)
80 GOTO 40
```

ベーシックの便利な命令

プログラムを作るとき、命令文をいちいちキーボードで打ち込むっていうのはじつにめんどくさい作業だよね。でも、RUNなどひんぱんに使用する命令は、そんなことしなくても平気な方法があるってことをキミは知ってたかな。ゲームベーシック画面からベーシック面を選んだときには、ファンクションキーが8つの命令の代役をしてくれるんだ。これなら1文字ずつ入力するわずらわしさがはぶけるね！

```
LIST
OK

RUN
OK
```

ファンクションキーの内容は？

ファンクションキーは、ゲームベーシックモード画面からベーシック面を選んだときには、F1。LOAD(M)、F2．PRINT、F3．GOTO、F4．CHR$(，F5．SPRITE、F6．CONT(M)、F7．LIST(M)、F8．RUN(M)というふうに定義されている。(M)はRETURNのことだ。どのキーがどの命令を代用しているか知りたくなったらKEYLISTと入力。すぐに画面表示するぞ。

```
KEY 1,"LOAD(M)
OK
KEY 2,"PRINT(M)
OK
```

ファンクションキーの利用法

F1からF8までのファンクションキーが8つの命令の代役をそれぞれはたしていることはもうおぼえたよね。でも、人によってはもっと別の命令がファンクションキーに入っていたほうがいいと思うかもしれない。そんなときは、KEY命令を使おう。ファンクションキー番号と命令の文字列を入力してやれば、どんな命令でもファンクションキーに定義することができるんだ。

```
F1 LOAD(M),F2 PRINT,F3 GOTO
F4 CHR$(,F5 SPRITE,F6 CONT(M
),F7 LIST(M),F8 RUN(M)
```

命令の省略形をおぼえよう！

ＰＯＳのような短い命令文ならいいけど、ＰＯＳＩＴＩＯＮのように字数が多くなると、１字ずつキーボードで打ちこんでいくのはかなりめんどうだ。ファンクションキーで代役させるのもいいけど、全部をそうするわけにもいかない。そこで、命令の省略形をおぼえよう。ほとんどの命令には省略形があって、たとえばＬＩＳＴならＬ.、ＩＮＰＵＴならＩ.ですんでしまうんだ。これを使ってラクラクプログラムだ！

```
LIST   -  L:
INPUT  -  I:
LOCATE-LOC:
```

残りのメモリを確認しよう！

長いプログラムを入力するときなんかは、先に進むにしたがって、残りのメモリエリアが気になってくるよね。

そんなときはＦＲＥ命令を活用しよう。ＦＲＥと入力するだけで、まだ使っていないユーザーメモリのバイト数が画面に表示されるぞ。プログラムが何行か進んだら、ＦＲＥ命令を使うようにする。こうすれば、残りのメモリエリアを気にしなくてもすむぞ。

```
PRINT FRE
 1230
OK
```

コントロールコード①

キーボードのＣＴＲコントロールキーにはさまざまな便利な使い方があるからおぼえておこう。たとえば、ＣＴＲキーを押しながらＡキーを押すと、ＩＮＳモードのＯＮ／ＯＦＦスイッチの役割りをはたし、Ｃキーと併用すると、ＢＲＥＡＫとなる。またＤキーと組み合わせれば、ＳＰＲＩＴＥ ＯＦＦ ＣＧＥＮ２、ＣＴＲ＋Ａの解除のほか、カラーパレットをバックグラウンド用のパレットコード１にすることができるぞ。

```
CTR+A  INSモード ノ ON/OFF
 :  +C  BREAK
 :  +D  ショキセッテイ CGEN2,SPRITE
        OFF,CTR+A ノ カイジョ
```

コントロールコード②

ＣＴＲキーをＥキーと組み合わせれば、カーソル以降１行分の消去ができ、Ｇキーと共に押せばＢＥＥＰ音を出す。Ｈキーと併用すればＤＥＬと同じ働きをするし、Ｊとならば▼キーと同じく行送りをする。また、Ｋキーとならカーソルをホーム位置に戻し、Ｌキーとなら画面をクリアする。Ｍキーとでは１行入力して改行、Ｒキーとでは ＩＮＳと同じ働きをする。ＣＴＲキーをうまく使いこなそう。

```
CTR+E  カーソル イコウ 1ギョウブン ショウキョ
 :  +G  BEEP
 :  +H  DEL ト ドウヨウ
 :  +J  ラインフィールド
 :  +K  カーソル ヲ ホーム ヘ モドス
```

Ｖ３の便利な命令①

ファミリーベーシックＶ３には、いままでになかった便利な命令が20種以上も付け加えられたぞ。たとえば、ＡＵＴＯ命令では、入力するプログラムの行番号を自動的に付けてもらえるし、ＤＥＬＥＴＥ命令を使えば、行番号を指定するだけで、プログラム中の消したい部分をまとめて消去することができるんだ。プログラムの行番号をつけかえるＲＥＮＵＭ命令もあるぞ。

BGTOOL , TRON , TROFF

Ｖ３の便利な命令②

Ｖ３では、モードを移行させる命令も追加されている。

たとえば、ＢＧＴＯＯＬ命令を使えば、ベーシックモードからＢＧグラフィックモードへすぐに移行できるんだ。また、ＴＲＯＮ／ＴＲＯＦＦ命令によって、トレースモードを簡単に実行・解除することができるぞ。

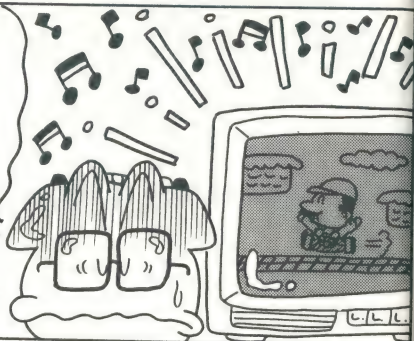
ほかにもＶ３では便利な命令がいっぱいある。他のページでもいくつか紹介してあるから参考にしてくれ。

AUTO , DELETE
RENUM

ベーシックのベーシック

ミュージック

この項目で出てくる命令語

PLAY

PLAY命令というのは　ゲームに使われとる音楽や効果音などに必要な機能のことです…
つまり　これには音楽的センスが必要で…
なにっ!?バカにするな!!
ああ〜〜この人に音楽のセンスがあるとは思えへん!!もうおしまいや!!
じまんじゃないがオレは音楽は
1だ!!
むん!!

わあ〜〜っ
やっぱり
ダメだ〜〜!!

かかかか

心配するな
オレは天才だから
少々のハンディなど
何でもない!!早く
説明をつづけなさい

はぁ　そんじゃ…
ムダやと思うけど…
ＰＬＡＹ命令には基本の
音階・音調のほかに
エンベロープやデューティ
効果までつけられ
ますのや

PLAY

デューティ効果
音色のこと

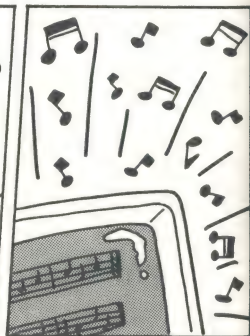
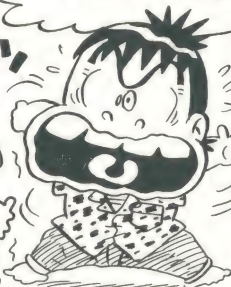
Y.M.O
それに　三和音まで
できて　ちょいとした
シンセサイザー
なみの機能や!!
PLAY
ピュ～イ
ピュ～イ
ピコピコ～ン
ほな、わては
荷物でもまとめ
させてもらいます…
カシャ
カシャ
うん　おもしろそうだ
やってみよ――!!
そうだっか
おーい　てきとうに
やったら
へんなのができた
ぜ～～!!
そうでっか
ほな　めいどの
みやげに
聞かせて
もらいまひょ

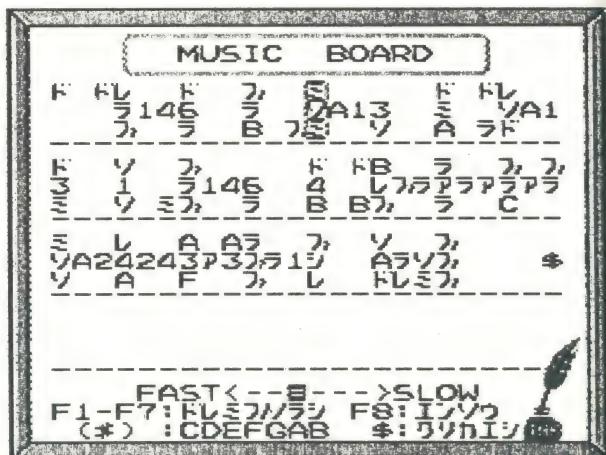
ずがーん

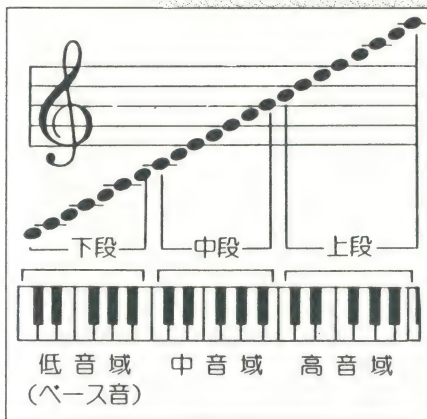
こ、これは
すごい
曲や!!

え——!?
なになに
あれ　よかったの!?
なー　おい
何とかいえよおい!!

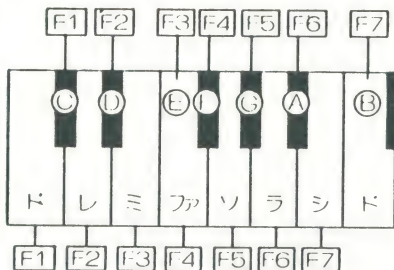
ホンマに　この
人はアホだか
天才なんだか
わからへん…

ヘターーッ

ミュージックボードでプレイ！

ファミリーベーシックで音を出すためのいちばん簡単な方法は、ミュージックボード画面を使うことだ。3音×24マス×4行の入力欄に、キミの好きな曲をドレミで入れていこう。入力方法も▲▼◀▶カーソルキーで、画面の羽根ペンを移動させ、そのつど、ドレミをファンクションキーで書いていくという簡単なもの。できあがれば、なん回でも自動演奏してくれるから楽しいぞ！

▲ミュージックボード画面を使って、自動演奏を楽しもう。どんな曲だってOKだぞ！

音域が広く、和音も出せるぞ!!

▲低音から高音まで3オクターブにわたって音を出せるぞ!

音階の入力欄は上、中、下の3段に分かれていて、それぞれに高音域、中音域、低音域(ベース音)を入力できるんだ。つまり3オクターブの音を出すことができるというわけだね。この3段を活用すれば和音も作れるぞ。

F1 F2 F3 F4 F5 F6 F7
C D E F G A B
ド レ ミ ファ ソ ラ シ ド
F1 F2 F3 F4 F5 F6 F7

◀ドレミはファンクションキーに対応しているんだ。シフトキーとファンクションキーを両方押せば半音が出るぞ

ベーシックで音を出してみよう

ベーシックで音を出すためには、PLAY文を使おう。この命令を実行させるためには、PLAY文の後に、音楽に関するいくつかのストリングデータを入力しなければならない。テンポ、デューティ効果(音色)、エンベロープ、オクターブ、音程、休符、音の長さ、重音をキミの好みに合わせて入力していくんだ。プログラムを実行させれば、コンピュータが自動演奏してくれるぞ。

```
LIST
10 PLAY "V15M1Y2O4"
20 PLAY "CDEFGFEDC"
OK
```

音　　程	指定方法
ド	C
ド#(レ♭)	#C
レ	D
レ#(ミ♭)	#D
ミ	E
ファ	F
ファ#(ソ♭)	#F
ソ	G
ソ#(ラ♭)	#G
ラ	A
ラ#(シ♭)	#A
シ	B

音の長さ		対応する整数
$\frac{1}{8}$(32分音符♪　)	𝅀	0
$\frac{1}{4}$(16分音符♪　)	𝄿	1
$\frac{3}{8}$(付点16分音符♪)	𝄿.	2
$\frac{1}{2}$(8分音符♪　)	𝄾	3
$\frac{3}{4}$(付点8分音符♪)	𝄾.	4
1(4分音符♩　)	𝄽	5
$1\frac{1}{2}$(付点4分音符♩)	𝄽.	6
2(2分音符♩　)	𝄼	7
3(付点2分音符♩)	𝄼.	8
4(全音符・　)	𝄻	9

和音を出してみよう！

和音というのは、ド・ミ・ソ、ミ・ラ・ドなどのように、2つ以上の音を同時に出す音のことだ。この和音を使うと、音にふくらみが出て、本格的な演奏もできるようになるからぜひためしてみよう。和音を演奏するには、〝C：E：G″のように各音の間にチャンネルセパレータ：(コロン)をはさんで入力する。音程は、ドをC、レをD、ミをE、ファをF、ソをG、ラをA、シをBと表記！

```
40 PLAY "C:D:E
50 PLAY "D:F:A
```

和音の各チャンネル

和音はすでに述べたように、〝チャンネルA：チャンネルB：チャンネルC″というふうになっている。このとき、各チャンネルごとに、音程はもちろん、音の長さ、テンポ、エンベロープ、デューティを指定する必要があるぞ。ただし、チャンネルCは、エンベロープ、デューティは変わらない。

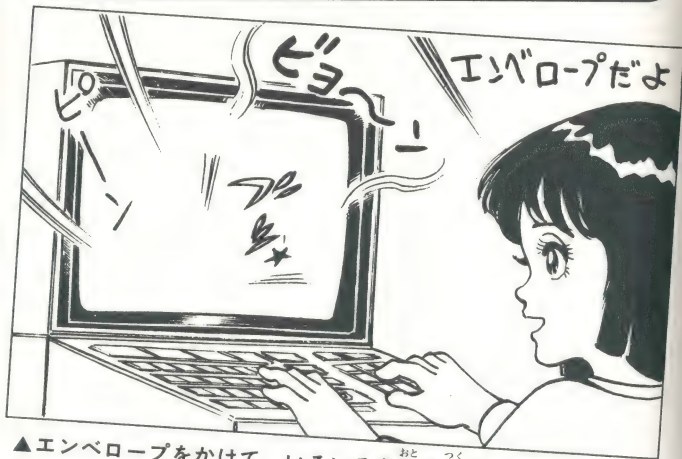
キミのベーシックにも和音をふんだんにとり入れて、幅のある音楽を作ろう！

```
10 PLAY "V15M1Y201
20 PLAY ":V15M1Y202
30 PLAY ":V15M1Y203
```

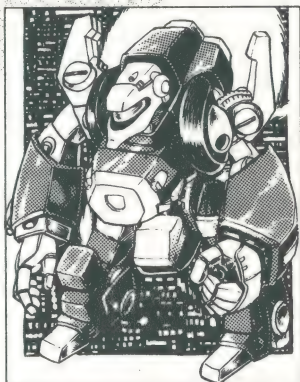
エンベロープを使ってみよう！

エンベロープをかけると、余いんのある音、切れた音、はねた音などが楽しめるぞ。

エンベロープをかけたいときは、M１と入力する。さらにその後に、Ｖを入力し、エンベロープの長さを指定。Ｖの後につく数字が少なければエンベロープは短くなり、大きければエンベロープは長くなる。ただしその範囲は０～15の間だ。エンベロープをかけないときはＭ０にしよう。

```
PLAY"M1V15C
PLAY"M1V10C
PLAY"M1V9C
```

▲エンベロープをかけて、いろいろな音を作ってみよう

サンプルを作ってみたぞ！

では、実際にベーシックを使ってプログラムした曲を紹介しよう。オリジナルビデオアニメ「メガゾーン23」の主題歌「背中ごしにセンチメンタル」だ。プログラムは、１番→間奏→２番のサビの部分というように入力されている。長いプログラムだけど、がんばって入力してみよう！

```
10 PLAY"T3M1V11Y2:T3M1V11Y2:T3M1
20 PLAY"R7:R3O2A1BO3CO2BAG:R7
30 PLAY"O2A8:R5O1AR:R5O0AR
40 PLAY"O3E3DRDCR:ARG:ARG
50 PLAY"O2B5G3A:RG:RG
60 PLAY"A6G3:RF:RF
70 PLAY"A5O3E3D:RF:RF
80 PLAY"RDCRO2B5:RGR:RGR
90 PLAY"G3ARA1B:GR:GR
100 PLAY"O3CO2BAGA5:AR:AR
110 PLAY"O3E3DRDCR:ARG:ARG
120 PLAY"O2B3O3C5D3:O2G3A5B3:RG
130 PLAY"E9:R5O1AO2CE3A:RFRF
140 PLAY"R:RO1FEFGFEF:R3FEFGFEF
150 PLAY"O2E6#G:#G6B:O1EEEEEE
160 PLAY"B5:R5:EE
170 PLAY"R3O3F5F3:R3O2#G5#G3:R3E5E3
180 PLAY"E5R3O2E:O1E5R:E5R
185 FOR A=0 TO 1
```

```
190 PLAY"EEEE:RO1A:O0A3ACA
200 PLAY"EDCDEAAA:RARE:AAEAAACA
210 PLAY"AGGGFF:RER:AACAO1DD
220 PLAY"FFFEDF:ARA:DDDDDD
230 PLAY"F8R3D:RARA:DDDDDDDD
240 PLAY"DDDDDC:RGR:O0GGBGGG
250 PLAY"O1BO2CDGGG:GRG:BGGGBG
260 PLAY"G7C3CC5:RGE3E6:GGBGO1CC6
270 PLAY"D3DD5E:#F3#F6R5:D3D6E3E
280 PLAY"F3E7R3:#GR#G:BEEEBE
290 PLAY"R7:O3E3DFE:O1EDFE
300 PLAY"R6E3EE:RG5A5R3:RG5A3O0AA
310 PLAY"EE:O1A5:CA
320 PLAY"EDCD:RA:AAEA
330 PLAY"EAAAAGGG:RERE:AACAAACA
340 PLAY"FFFFFE:RAR:O1DDCDDD
350 PLAY"DFF8:ARAR:CDDDCDDD
360 PLAY"R5E8:AR#GR:CDEEBEEE
370 PLAY"R3DE7:#GR#G:BEEEBE
380 PLAY"E3B:R:EE
390 PLAY"#GA9R3:ERERE:BEO0AACAAACA
400 PLAY"O0A5O2AAA:E9:O1C9
410 PLAY"AG3G:O0G6O1G3:R7
420 PLAY"RAAA:G6O0G3:R7
430 PLAY"A5G3G:G6O1G3:R7
440 PLAY"RFFF:G6O0G3:R7
450 PLAY"F6E3:O1C6O2C3:R7
460 PLAY"E7:C6O1C3:R7
470 PLAY"R7:C6O2C3:R7
480 PLAY"R7:C6O1C3:R7
490 PLAY"F3FFF:O2DDDD:O0D3DDD
500 PLAY"FFFF:DDDD:DDDD
510 PLAY"#F#F:#D#D:#D#D
520 PLAY"#F#F:#D#D:#D#D
530 PLAY"#F#F:#D#D:#D#D
540 PLAY"#F#G7R3:#D#DRO2#GAB:#D#DRO1EEE
550 PLAY"R8:A#GBA#G5:EEEEE5
560 PLAY"R5E7BO3C:R#G7#G#G:RE7EE
```

```
565 FOR B=0 TO 1
570 PLAY"O2B3AAO3C:O1E7:O0A3O1EO0EO1E
580 PLAY"RO2EBO3C:E:O0AO1EO0EO1E
590 PLAY"O2BA:E5:O0AO1E
600 PLAY"AO3C7R3:RE7:O0EO1EO0AO1EO0EO1E
610 PLAY"C7O2B:FF:DAO0AO1ADAO0AO1A
620 PLAY"O3C3O2B:F5:DA
630 PLAY"RF7R3:RF7:O0AO1ADAO0AO1A
635 IF B=1 THEN 720
640 PLAY"GGGG:D:O0GO1GO0GO1G
650 PLAY"GGGG:D:O0GO1GO0GO1G
660 PLAY"BBBB:G:O0GO1GO0GO1G
670 PLAY"BBBB:G:O0GO1GO0GO1G
680 PLAY"O3CCR5:G:CO2CO1CO2C
690 PLAY"O3C3CR5:G:CO2CO1CO2C
700 PLAY"O2B3BO3CO2B7:#G9:O1EE5E3EE5E3
710 PLAY"R5EBO3C:E5R#G7:O0E5RE7
720 NEXT
730 PLAY"GGGG7R3:D3D5D7R3:O0G3G5G7R3
740 PLAY"AAAA7R3:CC5C7R3:FF5F7R3
750 PLAY"BBBB7R3:DD5D7R3:GG5G7R3
760 PLAY"O3CCC7R3:AA5A7R3:FF5F7R3
770 PLAY"O2B6O3C3:#G6#G3:O1E6E3
780 PLAY"R7:R5E:R5O0E
790 PLAY"O2B6O3C3:#G6#G3:O1E6E3
800 PLAY"R7:R5E:R5O0E
810 PLAY"C3CCCCC:F8:F3O1FO0FO1FO0FO1F
820 PLAY"CC:R5:O0FO1F
830 PLAY"O2B7B5:#G#GR:O0E5ER
835 IF A=1 THEN 1060
840 PLAY"#G3A8R3:RC8:RA3O1EO0EO1EO0AO1E
850 PLAY"R5:R5:O0EO1E
860 PLAY"R7:R7:O0AO1EO0EO1E
870 PLAY"R3A1BO3CO2BAG:R:O0A5R
880 PLAY"A8O3E3D:R5ARA:RARA
890 PLAY"RDCRO2BR:RGR:RGR
900 PLAY"GA7G3:GRF:GRF
910 PLAY"A5O3E3D:RF:RF
```

```
920 PLAY"RDCRO2BRGA:RGRG:RGRG
930 PLAY"RO3FERDR:R8:O1D3AFADA
940 PLAY"CO2B7R3:R3O2F7R3:FAO0GO1GDG
950 PLAY"R7R3O3EDR:R9:O0GO1GDGCO2CO1GO2C
960 PLAY"CR:R5:O1CO2C
970 PLAY"O2BA7R3:R3C7R3:O1GO2CO0FO1FCF
980 PLAY"R7:R7:O0FO1FCF
990 PLAY"R3O3DCR:R7:O0#AO1#AF#A
1000 PLAY"O2#AR:R5:O0#AO1#A
1010 PLAY"O3CO2B5B3AR:R8:FBO0EO1EO0BO1E
1020 PLAY"#G5A:R7:O0EO1EO0BO1E
1030 PLAY"E6#GB5:O1#G6BO2D5:O0EEEEEEEE
1040 PLAY"R3O3F5F3E5:R3#G5#G3B5:RE5E3G5
1050 PLAY"R3O2E:R:R
1060 NEXT
1070 PLAY"#G3A6:RR3O2A1B:RO0A3O1A
1080 PLAY"R7:O3CO2BAGA5:O0AO1AO0AO1A
1090 PLAY"R7:O3E3DRD:O0AO1AO0GO1G
1100 PLAY"R5:CR:O0GO1G
1105 PLAY"O2B3O3C5D3:R7:O0GO1GO0GO1G
1110 PLAY"E7:R5O2A:O0FO1FO0FO1F
1120 PLAY"R:O3CE3D:O0FO1FO0FO1F
1130 PLAY"R3O3G1ABO4CDE:R7:O0GO1GO0GO1G
1140 PLAY"R3O3B1O4CDEFG:R7:O0GO1GO0GO1G
1150 PLAY"R7:R3O2A1BO3CO2BAG:O0AO1AO0AO1A
1160 PLAY"R7:A5O3E3D:O0AO1AO0AO1A
1170 PLAY"R:RDCR:O0GO1GO0GO1G
1180 PLAY"R:O2BO3C5G3:O0GO1GO0GO1G
1190 PLAY"O3E7:R5O2A:O0FO1FO0FO1F
1200 PLAY"R:O3CE3A:O0FO1FO0FO1F
1210 PLAY"R3O2FEF:R3O1FEF:R3O0FEF
1220 PLAY"GFEF:GFEF:GFEF
1230 PLAY"REER:RBBR:RO1#GGR
1240 PLAY"FFRG:O2CCRD:AARB
1250 PLAY"GR#G#G:DRDD:BRBB
1260 PLAY"RA1BO3CO2BAG:R7:R7
1270 PLAY"A5O3A:R5O3E:R5O2C
1280 PLAY"R3G5AR3:R3D5ER3:R3O1B5O2CR3
```

PLAY文でゲームの音を出す

PLAY文はなにも曲を演奏させるためだけのものじゃない。ゲームには欠かせない効果音を作る役割りもはたしてくれるんだ。

効果音の作り方は基本的には、音楽の入力方法と同じ。いくつかゲームに必要な効果音のリストをあげておくから、それを参考にして、キミも入力してみよう。ゲームを作るのも、ゲームで遊ぶのもよりいっそう楽しくなるゾ。

```
PLAY
```

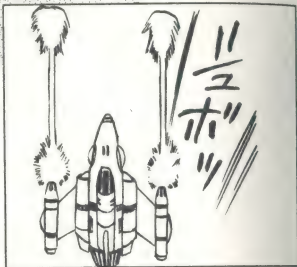
ゲームのスタートの音

ゲームのスタート時には、やっぱりなにか開始を告げるサウンドが欲しいよね。そこで、下のリストのような音をプログラムしてみたゾ。

どうだい、いかにもゲームスタートにふさわしいだろう。


```
10 PLAY"M1V9Y2T3:M1V7Y1T3:M1T3"
20 PLAY"O3E5G:O3C5C:O2G5G"
30 PLAY"O4CO3B4A1:CC:#F#F"
40 PLAY"G4A1:F4F1:O3D4O2B1"
50 PLAY"B4O4C7:F4E7:G4C7"
```

ミサイル発射音を作ろう！

戦争形式のゲームでは、ミサイルは登場キャラクタの主力武器。ここはやっぱり迫力あるミサイルの発射音が欲しいところだ。そのリストを下にあげておいたから、キミも参考にしてくれ！

```
10 PLAY   "V15M1Y0T101"
20 PLAY   ":V15M1Y0T101"
30 PLAY  "::V15M1Y0T101"
40 PLAY  "G1FEDC:A1GFE:B1AGFE
"
```

爆発音を作るぞ！

ミサイルが命中したのなら、こんどは爆発音も欲しい。下のプログラムは、爆発音のリストだ。

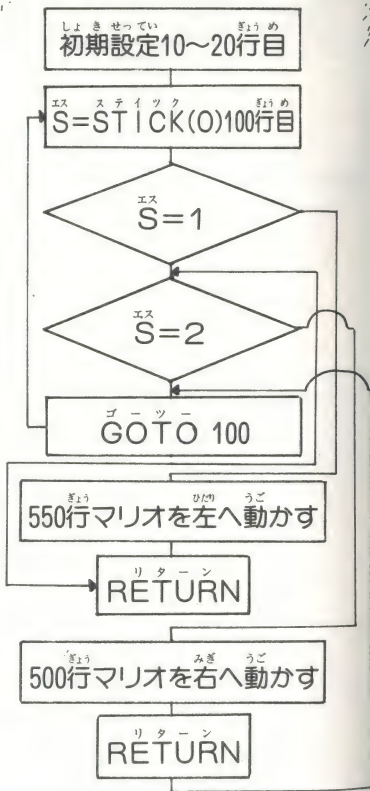
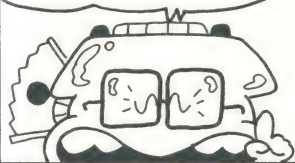
まだこのほかにも、サウンド機能を使えば、いろんな効果音を作り出せるぞ！

```
10 PLAY   "V15M1Y0T104"
20 PLAY   ":V15M1Y0T104"
30 PLAY  "::V15M1Y0T104"
40 PLAY  "C1EFGA:D1CEF:F1CEFB
"
```

ベーシックのベーシック
まとめ

この項目で出てくる命令語

INPUT　GOTO　GOSUB　LOCATE
PRINT　STICK　DEFMOVE

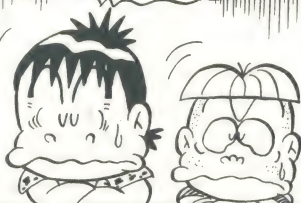
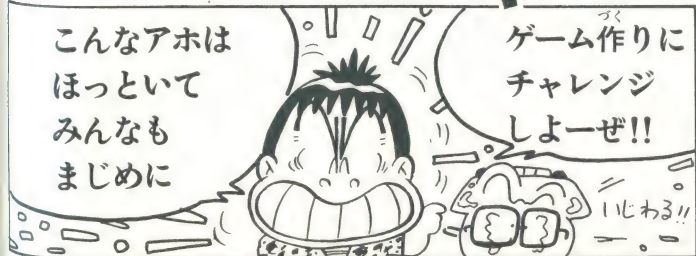
プログラムはある一定の流れをもっててそれを図にあらわしたのをチャートというねん!!
初期設定10〜20行目
S=STICK(0)100行目
S=1
S=2
GOTO 100
550行マリオを左へ動かす
RETURN
500行マリオを右へ動かす
RETURN
うへ〜なんじゃいこりゃ!!

それが　しらずしらずの
うちに流れになっていくんや
プログラムは命令が
集まったもんやけど
それぞれが　特長をもった
プログラムなんや!!

なんか えらく
はりきってんな
こいつ…

もーおわり
だからじゃ
ないですか

プログラムを
作っていけば
自分のプログラムに

くせがあることが
わかってくるはずや!!

それには
流れを
つかむことが
大切なん
や――!!
そいつが個性なんや!!
プログラムに個性を
もたせなければ
オリジナルゲーム
とは
いえんのや!!
ずっぱ～～ん!!
それじゃ
流れをつかみに
いこ～っと
?
そおかぁ～～
流れが大切
なのか～～～

んモ〜〜〜っ
すぐそーやって
みえみえの　ギャグ
やるんだから〜〜
もういや――――!!
どーか　うまく
流れます
よーに!!
ドンブラコ
ドンブラコ
ゲーム作りに
チャレンジ
しよーぜ!!
こんなアホは
ほっといて
みんなも
まじめに
いじわる!!

大きなプログラムを作るには!?

1本のゲームを作ることは、これからベーシックを覚えようとしていたり、初めてゲームのプログラムを作るキミにとっては、どんなに小さなプログラムでも大きなプログラムに感じるだろう。プログラムを作る時に必要なのは、プログラムの流れを考えることで、一番いいのは人のプログラムを読むことだ。

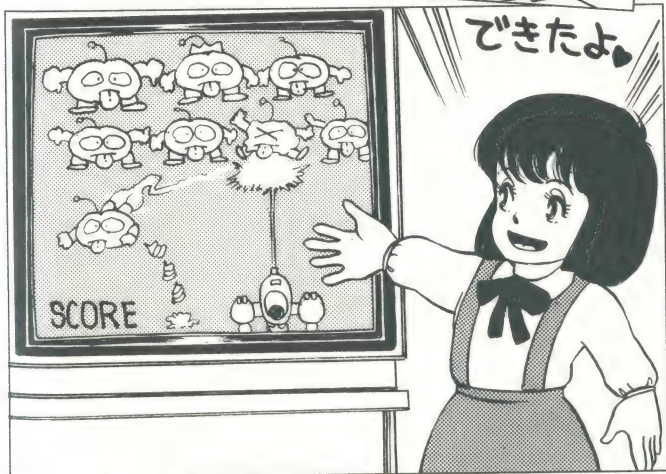
アイディアを出そう!

人のプログラムをプレイしている時に"こうしたらいいのになっ"て、思うことがあるだろう。そんな時にそのアイディアをメモしておくんだ。そして、ベーシックの命令がわかる様になってきたら、まず人のプログラムをアイディア通りに改良してみよう。それを、くりかえすとプログラムがよくわかってくるはずだ。

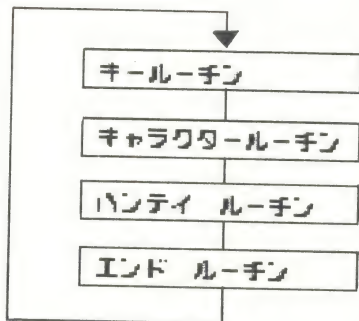
オリジナルゲームを考える

人のプログラムを思った通りに改造出来る様になったら、今度は自分だけのオリジナルゲームを考えてみよう。

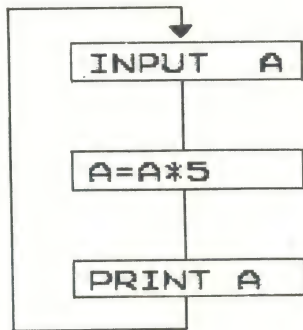
初めは、人のゲームのまねごとでもいいから色々アイディアを、出して見るんだ。

そして、今の自分に一番出来そうなゲームをえらび、どうすれば、考えた通りのプログラムが出来るかを考えるんだ。この時に、人のプログラムが参考になる。

プログラムの流れを考える

オリジナルゲームを作るためのルーチンが出そろったら、今度はそれを組み立てるんだ。プログラムを組み立てると言っても、ただならべて、出来上りと言うわけにはいかない。メインルーチンとかキーの入力ルーチンとかキャラクターの出力ルーチンとかに分けて、構造的に組み立てなければならないんだ。下の図の様な大ざっぱなプログラムの図をチャートと言って、プログラムの流れを図にしたものなんだ。

```
10 REM フロチャート
 :
 :
 :
```

色々なことをチャートにしよう

チャートを書くには、まずどんなことをするのか、目的をはっきりさせなければならないんだ。下の図は、数字をキーボードから入力してからその数に5をかけ算したのを画面に出してくり返して同じ処理をするチャートだけど、この場合の目的である様に入力した数を5倍にして画面に出すと言うことがチャートから分るはずだ。ゲームでは、主にこの様なはっきりした目的がない場合が多いから、注意して作ってくれ。

```
10 REM ケイサン チャート
 ・
 ・
 ・
```

チャートをプログラムにする

チャートが出来たら今度は、それをプログラムしなければならない。下のプログラムは前のページのチャートをそのままプログラムにしてみたものだが、こんな感じでチャートがプログラムになるんだと言うことが、わかってもらえればいい。本当に大きいプログラムになると、どうしてもサブルーチンが多くなって、プログラムの流れの中心であるメインプログラムが重要になってくるんだ。

```
10 プログラム チャート
 :
 :
```

```
10 INPUT  A
20 A=A*5
30 PRINT A
40 GOTO 10
```

メインプログラムとは?

サブルーチンは、GOSUBで実行する時がほとんどで、メインルーチンと呼ばれるプログラムの流れの中心と言うべきルーチンからコールされるんだ。今のゲームのほとんどが、メインルーチンを変えることによって色々変化をつけたり、まったくほかのプログラムに作り変えたりすることが出来るんだ。又、自分のオリジナルゲームでほかのプログラムを作る時にも大きく変わるのが、メインルーチンだ。

```
10 REM メイン ルーチン
 :
 :
```

```
10 REM メイン ルーチン
20 GOSUB 100
30 GOSUB 200
40 GOSUB 300
50 END
100 REM INPUT
110 INPUT " A = ",A
120 RETURN
200 REM ケイサン
210 A=A*5
220 RETURN
300 REM PRINT
310 PRINT A
320 RETURN
```

オリジナルゲームを作る

例として、これからオリジナルゲームを作ってみよう。まずマリオをスティックの方向どおりに動かすサブルーチンで、1020行でスティック命令で変数Kにキーの押された方向のデータを入れて、1030行で方向をＩＦ文で判定して、マリオの動く方向を変数に入れている。1040行はマリオを画面上で動かす命令なんだ。

```
10 スティック メイレイ(サブルーチン)
  :
  :
```

```
1000 M=0
1010 IF MOVE(0) THEN RETURN
1020 I=STICK(0)+1
1030 K=VAL(MID$(".37.546.128
....",I,1))
1040 DEF MOVE(0)=SPRITE(0.K,
1,4,1)
1050 MOVE 0
1060 RETURN
```

オリジナルゲームを作る

つぎに、リンゴをマリオが取るゲームにしたいから、りんごを画面に出さなければならない。下はそのリストで、2000行で画面を消して、2010行でＦＯＲ命令で変数Ｙが、22になるまでくり返す様にする。2020行でＸに１～26の乱数を出させる様にして、1040行でＬＯＣＡＴＥのよこ座表にＸ、たて座表にＹを使ってやり、2030行でＣＨＲ＄命令でリンゴをプリントしてやっているんだ。

```
10 FOR メイレイ
 : ヘンスウ Y,ランスウ X
 :
```

```
2000 CLS
2005 L=0
2010 FOR Y=1 TO 22
2020 LOCATE RND(25)+1,Y
2030 PRINT CHR$(215);
2035 L=L+1
2040 NEXT
2050 RETURN
```

オリジナルゲームを作る

つぎに、マリオにりんごを取れたか、取れなかったかを、判定するサブルーチンを作ってみる。

3020、3030行では、スプライトの座表とキャラクタの座表とが合わないので、うまく合う様に計算してXはX1、YはY1の変数に入力してやっている。

3040行でＳＣＲ＄でX1、Y1の数値でもとめられるキャラクタを、判定しているんだ。

```
10 ハンテイ サブルーチン
  :
  :
```

```
3000 X=XPOS(0)
3010 Y=YPOS(0)
3020 X1=(X- 8)/8
3030 Y1=(Y-20)/8
3035 IF X1<0 OR X1>27 THEN R
ETURN
3036 IF Y1<0 OR Y1>23 THEN R
ETURN
3040 IF SCR$(X1,Y1)<>CHR$(21
5) THEN RETURN
3050 S=S+1
3060 L=L-1
3070 LOCATE X1,Y1:PRINT " ";

3080 PLAY "CDE:EGA"
3090 RETURN
OK.
■
```

オリジナルゲームを作る

つぎは、ゲームを面白くするためにどのぐらいの時間で全部取ることが出来たかタイマーのルーチンと、ゲームオーバーの表示のサブルーチンを作ってみよう。

まず、310行でりんごが全部取れたかどうか判定しておいて、全部取ることが出来たら400行へ、取れなければ200行へ行く様にする。420行はタイマーのカウントと表示で、440行はゲームオーバーの表示。

```
10 タイマー ルーチン
 : ゲームオーバー ルーチン
```

```
400 LOCATE 0,0
410 PRINT "TIME:";
420 PRINT T
430 LOCATE 10,10
440 PRINT "GAME OVER"
450 IF T<=B THEN 600
460 GOTO 900
600 LOCATE 7,12
610 PRINT "YOU GET HISCORE"
620 B=T
900 IF STRIG(0)<>1 THEN 900
910 GOTO 10
990 END
OK
```

オリジナルゲームを作る

さて、各サブルーチンは出来ました。つぎにこのプログラムをまとめるプログラム、つまりメインプログラムを作りましょう。1行から、40行までは、プログラムの初期設定をして、120行のGUSUBでりんごを出すルーチンにコールさせて、200行でキーの対応ルーチンに、300行がタイマー用ルーチン、310行でメインの200行へくりかえしの判定をしているんだ。このプログラムは全部入力するとちゃんと動くゾ。

```
200 GOSUB 1000
210 GOSUB 3000
300 T=T+1
310 IF L THEN 200
```

```
1 B=999
10 CLS
20 SPRITE ON
30 POSITION 0,120,125
40 CGSET 1,1
50 PLAY "V15M1Y104C1"
60 PLAY ":V15M1Y101C1"
70 PALETB 0,9,48,22,24
100 T=0
110 S=0
120 GOSUB 2000
130 LOCATE 14,0
140 PRINT "BEST:";
150 PRINT B
160 PLAY "CDEFG:DEGAB"

■
```

ベーシックのベーシック

エラーメッセージ

この項目で出てくる命令語

NF	OM	ST	NR
SN	UL	FT	RE
RG	SO	CC	NP
OD	DD	UF	UP
IL	DZ	MO	
OV	TM	TP	

ＮＦ　NEXT without FOR

ＦＯＲ～ＴＯ～ＳＴＥＰ／ＮＥＸＴ文で、ＦＯＲがないのにＮＥＸＴがあるときはＮＦエラーとなる。ＮＥＸＴの前には、必ずそれに対応するＦＯＲが入っていなければならないんだ。ＦＯＲがぬけ落ちていないか、よく確かめて、入力しなおそう。

ＳＮ　Syntax error

ＳＮのエラー表示が出た場合は、文法がまちがっているということだ。もう一度、文法を見なおして、まちがえた箇所を訂正しよう。特に、１とＩ、０とＯ、・と，などはまちがいやすいから注意しよう。また、文字と文字のスペースにも気をつけること。

ＲＧ　RETURN without GOSUB

ＧＯＳＵＢがないのにＲＥＴＵＲＮがあるときは、ＲＧのエラー表示が出るぞ。ＧＯＳＵＢはサブルーチンを呼び出すための命令で、ＲＥＴＵＲＮはサブルーチンを戻り先に復帰させるときに使う命令。ＧＯＳＵＢとＲＥＴＵＲＮは必ず一組として考えよう。

ＯＤ　Out of DATA

ＯＤのエラー表示が出るのは、ＲＥＡＤで読むべきデータが、ＤＡＴＡ文に用意されていないからだ。ＲＥＡＤ文とＤＡＴＡ文もつねに対にして使うステートメントで、どちらが欠けていても、プログラムは実行されない。ＤＡＴＡ文の中にはデータとなる定数を書き入れよう。

ＩＬ　Illegal function call

ステートメントや関数の呼び方がちがっているとこのエラー表示が出るぞ。ＳＰＲＩＴＥ命令などの変数や数値のまちがいが考えられる。まちがえた箇所を探すのは大変やっかいな作業だ。どうしてもみつからないときは、リストを最初から見なおすしかない。

ＯＶ　Overflow

計算をする時に、演算結果が許容範囲を越えたとき、ＯＶのエラー表示が出る。カリキュレータボード画面の許容範囲は８ケタまで。計算の答が８ケタを越えると、答は表示されずに、エラーとなる。自分で計算して確かめてみよう。

オーエム OM
Out of memory

メモリが不足している。V1、V2を使っている人によく出るエラーだ。プログラム中のブランクが大きい場合や同じプログラムが2つ入力されている場合などにこのエラーが出る。大きなプログラムを作るときは、FRE命令で残りのメモリを見ながら入力していこう。

ユーエル UL
Undefined line Number

GOTO、GOSUB、IFなどで、指定した分岐先の行番号がないと、ULのエラー表示が出る。つまり、それらの命令によって分岐したプログラムが、どこへ行っていいのかわからなくなってしまうためにおこるエラーだ。複雑なプログラムを作っても基本的なことを忘れちゃいけないよ。

エスオー SO
Subscript out of range

SOのエラー表示は、配列変数の添字が、規定外のときに出るんだ。リストを見なおしてチェックしよう。規定外の添字が入力されているはずだから、ちゃんと訂正しよう。

エラーというのはちょっとしたミスが原因ということが多い。気をつけよう。

DD　Duplicate Definition

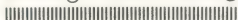
配列が２重に定義されているとＤＤのエラー表示が出るぞ。ＤＩＭ命令では、同じ変数を２度にわたっては定義できないんだ。同じ変数を２回呼び出していないか、あるいは、ＧＯＴＯ文によってそこを繰り返していないか、もう一度チェックしてみよう。

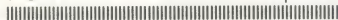
DZ　Division by zero

０によるわり算をした場合、このＤＺのエラー表示が出る。たとえば１÷０という計算。こういう計算はあり得ないものなので、エラーになるんだ。まちがえて、こんなわり算をしていないか、たしかめてみよう。これは算数の基本でもあるんだよ。

TM　Type mismatch

このＴＭのエラー表示が出るということは、変数の型が一致していないからだ。変数には数字型と文字型がある。文字型の場合は＄をつけるのが決まりだ。だから、Ａ＋Ａ＄という式は数字＋文字ということになって成り立たない。型のちがう変数はいっしょにしないということだ。

ST String too long

文字が31文字を越えてしまっているにもかかわらず、プログラムを実行しようとするとSTのエラー表示が出る。その部分の文字を31字以内に収めるようにしよう。決められた字数はきちんと守ることが大切だ。欲ばりすぎてエラーを出したのではもともこもないよ。

FT Formula Two complex

式が複雑すぎると、このFTのエラー表示が出るんだ。たとえば、(　)が異常に多い場合などが考えられる。いくらコンピュータといっても、あまり無理な式は入力しないこと。すっきりとした形に式をまとめあげることも、大切なテクニックの1つなんだよ。

CC Can't continue

CONTによってプログラムの実行を再開できないときは、このCCのエラー表示が出る。原因としては、ENDや他のエラーの発生によってプログラムが止まったり、CLEAR直後、あるいはプログラムを止めてからプログラムの書き換えをおこなった場合などが考えられる。

UF（ユーエフ）

Undefind function（アンダファインド ファアンクション）

これは、ファミリーベーシックV3に新たに追加されたエラーだ。未定義の関数を呼んだ場合、このUFのエラー表示が出る。まちがえた箇所、おかしな箇所はきちんと訂正して、時間のムダをはぶこう。コンピュータになれておくことが大切だ。

MO（エムオー）

Missing operand（ミッシング オペランド）

パラメータの必要な命令に指定がないとき、MOのエラー表示が出る。もう1度、ゆっくり見なおして、指定を入れ忘れていないかチェックしよう。ほんのちょっとしたことがエラーにもつながり、逆に、キミの勉強にもなってベーシックになれていくんだ。

TP（ティーピー）

Tape read ERROR（テープ リード エラー）

このTPのエラー表示が出たということは、カセットテープからデータが正しく読みとれていないということだ。カセットはきちんと入れてあるか、接続はおかしくないか、いろいろと調査してみよう。

NR

No RESUME

ON ERROR GOTOで、分岐先にRESUME文がないとNRのエラー表示が出るぞ。ルーチンの終了行をはっきりさせておこう。

RE

RESUME without error

RESUME文のあるルーチンへ、ON ERROR GOTO以外から分岐した場合、このREのエラー表示が出るんだ。

NB

No BG data

BGPUT命令を使う時、BGデータがRAM領域にない場合、このNBのエラー表示が出る。BGPUTで、RAM領域内にBGデータを送るようにしておこう。

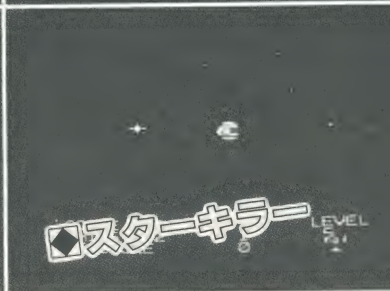
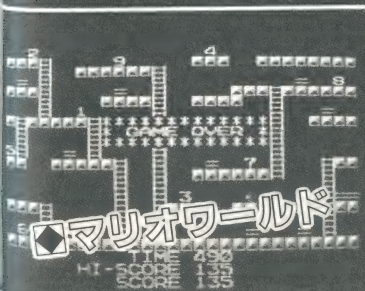
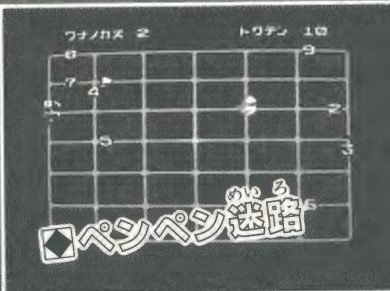
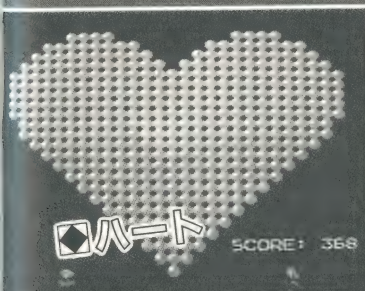
UP

Unprintable error

いままでのどれにもふくまれないエラーがあるとき、このUPのエラー表示が出る。それだけにまちがった箇所を探し出すのは大変だが、たんねんに調べよう。

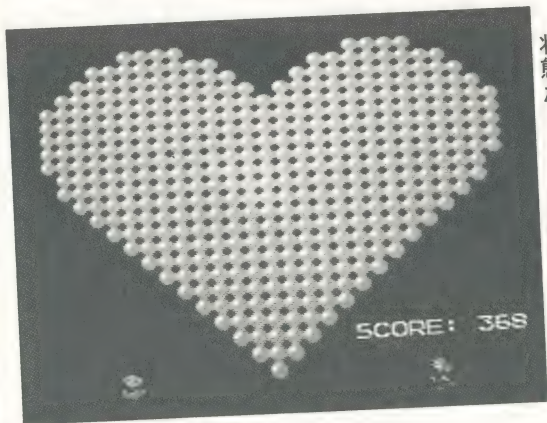
ファミリーベーシックV3

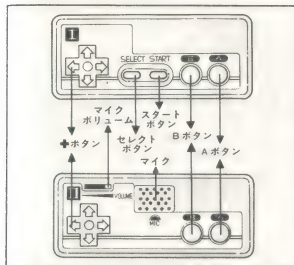
■内蔵プログラム■

ファミリーベーシックには4つのゲームが内蔵プログラムされているぞ。そのままでも遊べるけど、改造を加えて、より楽しいゲームにしよう！

ハート

キミの息や声をマイクに吹きかけて、画面にハートのマークを完成させよう。改造プログラムの作り方もそっと教えちゃうぞ！

■遊び方

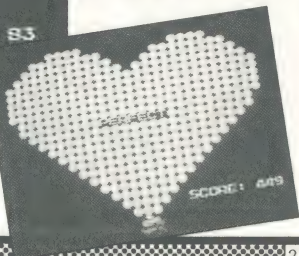
コントローラのマイクに息を吹きかけるか、声を出すかして、画面にハートのマークを描くゲームだ。プログラムを呼び出すときは、F1キーを押すか、GAME　0 RETURNとキーを押す。すると、PLAY　？　Y／N＝>とメッセージが出てくるぞ。遊びたいときは、Y RETURNとキーを押そう。これで画面は暗くなり、ハートを描く準備ができたんだ。

Nを選ぶと、画面にハートのマークが現われる。ゲーム完了時の状態だ。

■コントローラーの使い方

ハートを描くにはⅡコントローラーを使う。VOLUME(ボリューム)のつまみをいちばん右まで動かしてから、マイクの左上の小さな穴に向かって息を吹きかけるか、声を出すかするんだ。

■プレイしよう!!

マイクに息を吹きかけると、画面にハートのマークが少しずつ出てくるぞ。次にマリオとレディーが画面の両端から出てくる。ハートを完成させて2人をくっつけよう!

◀息をふきかけたり、声を出すたびにハートのマークは完成に近づいていく。マリオとレディーも近寄ってくるぞ!

▶ハートが完成し、マリオとレディーが真ん中で出会うと、音楽が鳴ってゲームは終了する

改造プログラムに挑戦しよう!!

ハートの色を変えてみよう!

ハートを描いている色は、プログラムの80行で決められているんだ。だからここに入力されている数字（Vの値）をほかの数字に変えれば、ハートの色も変えられるぞ。&Hというのは、16進数を意味している。

また、音楽が流れているときのハートの色は430行のWによって決められている。これも変えてみよう。

```
90 PALETB 0,15,48,48,V
480 LOCATE10,10:PRINT"PERFEC
T":FORN=0TO100:PALETB 0,6,&H
30,&H30,V:SWAPV,W:NEXT
```

バックの色を変えてみよう!

バックの色は最初黒だけど、これは、90行のＰＡＬＥＴＢ命令で変えることができるぞ。ただし、バックの色を変えるときは、ハートの色と同じにならないようにしよう。同じ色にするとハートが見えなくなってしまうぞ。また、音楽が流れているときのバックの色を決めているのは480行だ。

```
80 SPRITE ON:CGSET0,0:V=&H15
430 W=&H34:FORN=0TO1:DEFMOVE
(N)=SPRITE(N,3+4*N,1,1,0,0):
POSITIONN,236*N,200:NEXT
```

ハートを描くキャラクタを変えよう!

ファミリーベーシックの内蔵コンピュータでは、ハートを丸いボールで描いているけど、それはほかのキャラクタで描くことも可能だ。300行と350行、410行の"●"をH$に変えた後、135行を新たに追加しよう。その中のCHR$( )の数字を変えると、いろいろなキャラクタでハートを描くことができるぞ。184～255の数字だとBGグラフィックで使うキャラクタに、33～183だと文字や記号になるんだ。

```
300 LOCATEX,Y:PRINT"●";:NEXT
:RETURN
350 X=X+1:IFSCR$(X,Y)=" "AND
 X<=E THEN LOCATE X,Y:PRINT"
●":GOTO330
```

マイクに音が入っているか確かめよう

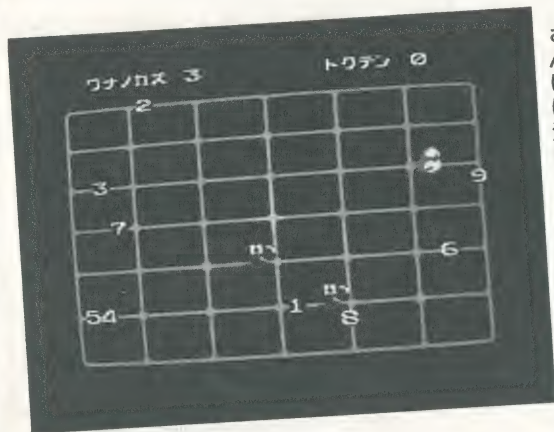
コントローラーのマイクに音や声が入っているかどうかを知るには、PEEK( )命令とメモリ&H4016番地を使おう。下のリスト通りのプログラムを入力してRUNすると、画面に次々と数字が出てくるから、VOLUMEをいっぱいにして、マイクに音声を入れよう。マイクに音がちゃんと入っていれば、画面の数字が変化するぞ。

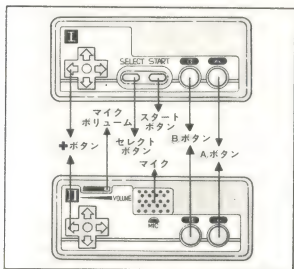
```
10 PRINT PEEK(&H4016)AND&H4;
20 GOTO 10
```

ペンペン迷路

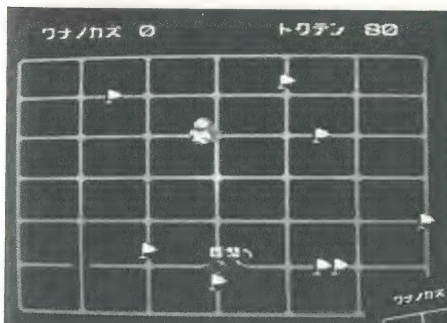
迷路にある数字をペンペンに通過させるゲームだ。改造すれば、敵の数を増やすこともできて、ゲームがいっそう楽しくなるぞ！

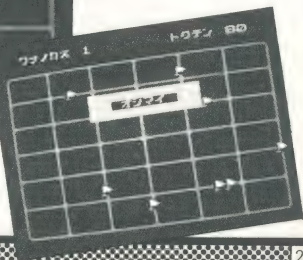
■遊び方

迷路にある１～９の数字を、ペンペンを操作して、小さい順に通過していこう。１個通過するごとに10点の得点になるぞ。だけど、迷路にはカニさんがいて、ペンペンを追いかけてくる。カニさんにつかまらないようにしよう。ゲームを始めるには、[F 2]キーか、[G][A][M][E]　[1][R E T U R N]を押してプログラムを呼び出し、コントローラー[I]のＡボタンを押せ。

◀迷路の中にある数字を小さい順に通過していこう。ただし、カニさんには注意

■コントローラーの使い方

〝ペンペン迷路〟では、コントローラーⅠを使うぞ。コントローラーの✚ボタンを押して、ペンペンを迷路にそって動かしていくんだ。おわったゲームを再開させるには、STARTボタンを押そう。

■プレイしよう‼

ペンペンはワナを3つ待っているぞ。Aボタンを押して迷路におくんだ。ワナにかかったカニさんは、しばらく動けなくなるから、そのすきに、数字をクリアしていこう。

◀迷路の数字をすべて通過しろ！　おいかけてくるカニさんには、ワナをしかけて時間をかせごう。ガンバレ！

▶ペンペンがカニさんにつかまってしまうとゲームオーバーだ。ウーム、残念！

改造プログラムに挑戦しよう!!

色を変えてみよう!

迷路の色や背景の色を変えたいときは、下のリストのようなプログラムを1行追加すればいいんだ。このときの数字が色を表わしているんだよ。

色を変えるとゲームをやめても、その色のままになっているから、もとに戻したいときは、CTRキーを押しながらDキーを押そう。

```
105 PALETB 0,10,48,19,37
```

カニさんの数を変えてみよう!

下のプログラム中のKNがカニさんの数を指定しているんだ。この数字を変えるとカニさんの数も変わるぞ。ただし、5つ以上のキャラクタを同時に横に並ばせると、5つめは画面から消えてしまうから注意。カニさんの数は1～3匹ということにしよう。

```
370 KN=2:W=219
```

タイマーを作ってみよう！

タイマーを作って、ゲームに制限時間を設けてみよう。タイマーを作るには、下のリスト通りのプログラムを追加すればいいんだ。

このタイマーによって、ある一定時間がたつとゲームはストップ、緊迫感がだんぜん盛りあがるぞ。

よりいっそうゲームを楽しくするために、いろんな工夫をしてみよう。

```
465 DEFMOVE(5)=SPRITE(13,3,4
5,128,0,0):POSITION5,116,210
466 DEF MOVE(4)=SPRITE(6,3,2
5,128,0,0):POSITION 4,16,210
:MOVE 5,4
475 IF CRASH(5)=4 THEN 1120
```

カニさんの動きを変えてみよう！

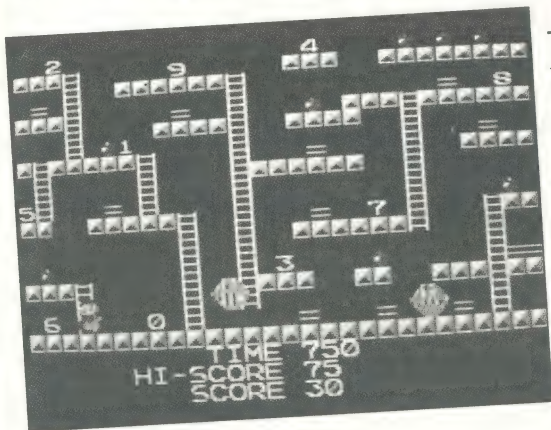
迷路が途切れている場所というのは、ペンペンは通れないのに、カニさんは通れるようになっているんだ。これでは不公平、カニさんもペンペンと同じように、迷路の切れ目を通れなくするためには、プログラムを下のリストのように変えればいいぞ。880行のＴＨＥＮの後の1060を940にすればいいんだ。

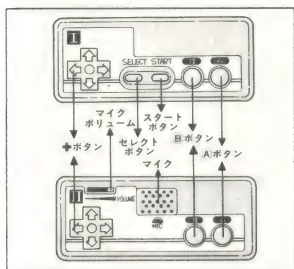
```
880 IF (KD=3 AND JX=1)OR(KD=
7 AND JX=-1)OR(KD=5 AND JY=1
)OR(KD=1 AND JY=-1) THEN1060
```

マリオワールド

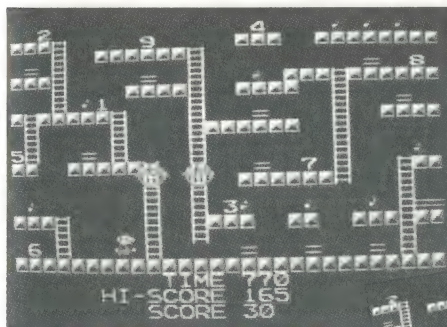
ＢＧグラフィックを使った本格的なゲームだ。マリオをコントロールして床の上の数字を、リンゴを取りまくろう。改造によっていちだんとグレードアップできるぞ。

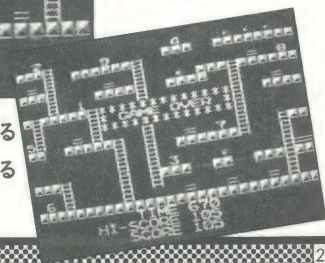
■遊び方

［Ｆ３］キーか［Ｇ］［Ａ］［Ｍ］［Ｅ］　［２］［，］［１］［ＲＥＴＵＲＮ］を押してプログラムを呼びだそう。コントローラーは１を使うぞ。ゲームの画面は下の写真の通り。床の上にある10個の数字と10個のリンゴをマリオに取らせるゲームだ。数字は１個30点、リンゴは１個15点になるぞ。ただし、数字は小さい順に取っていかないと得点にならない。マリオの敵はニタニタだ。ジャンプしよう。

ジャンプしてニタニタを飛び越えよう。ハシゴの使い方もうまくマスターしておこうね！

■コントローラーの使い方

✚ボタンでマリオを左右に歩かせよう。ジャンプするときはＡボタンを使う。そのとき✚ボタンを使えばジャンプする方向が決められるぞ。また、ハシゴを登るときは、✚ボタンを上に押すんだ。

■プレイしよう!!

このゲームには制限時間があるぞ。時間内に数字とリンゴを全部取らないとゲームオーバーだ。もちろん、ニタニタにつかまったら、ゲームはおしまいだよ。

◀床の上の「＝」はジャンプ台だ。マリオがここで立ち止まると、自動的にひとつ上の床に飛び乗ることができるぞ！

▶画面の下のＴＩＭＥが０になるか、マリオがニタニタにつかまるとゲームオーバーだ

改造プログラムに挑戦しよう!!

ニタニタの数を変えてみよう！

内蔵プログラムでは、ニタニタは２匹で追いかけてくるようになっているけど、その数を増やしたり減らしたりするには、プログラム中の１行を変えればいいんだ。下のリストのＮＮがニタニタの数。ただし、その数は実際、画面に登場しているニタニタの数より１つだけ少ないんだ。たとえばＮＮの数が１なら、ニタニタの数は２匹ということだよ。

```
140 T=800:W=0:NN=1:X=116
```

得点を変えてみたぞ！

数字は１個30点、リンゴは１個15点っていうことになっているけど、この得点を変えることだってできるぞ。

数字は420行、リンゴは880行のそれぞれＷ＝Ｗ＋の後の数を変えればいいんだ。ただし数字とリンゴの得点の合計が32767を越えるとエラーになるぞ。

```
420 IF ASC(SCR$(XL,YL-1))=Z
THEN LOCATE XL,YL-1:PRINT "
":Z=Z+1:W=W+30:LOCATE 14,23:
PRINT W;
```

マリオの最初の位置を変えてみよう!

最初にマリオが画面に登場する位置は、140行のXとYによって決められているんだ。だからこの数字を変えれば、マリオの登場位置を変えることができるぞ。

X(ヨコ)は16～224、Y(タテ)は16～168くらいの間で選ぶのがちょうどいいよ。

マリオの位置を変えれば、コースのまわり方も違ってきて、同じ画面でも新鮮な感じがするぞ。

```
140 T=800:W=0:NN=1:X=116:Y=1
27:POSITION 7,X,Y:GOSUB 890:
V=0:GOTO 540
```

ジャンプ台でのジャンプ力に変化を!

マリオはジャンプ台の上に乗ると、自動的にひとつ上の床に飛び乗れるようになっている。このとき、ジャンプできる高さをいろいろと変えてみると面白いよ。ジャンプの高さを決めているのは、リスト中のGの数だ。数の絶対値を大きくすればするほどマリオは高く飛び上がり、少なくすればマリオのジャンプ力は弱まるよ。

```
410 IF SCR$(XL,YL-1)=K$ THEN
 G=-15:GOTO 770
```

スターキラー

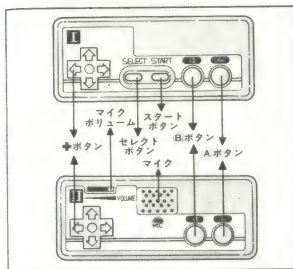
宇宙を舞台にしたゲームだぞ。１人でも２人でも遊べるんだ。改造を加えればいっそうグレードアップ。友だちとハイスコアを競おう！

■遊び方

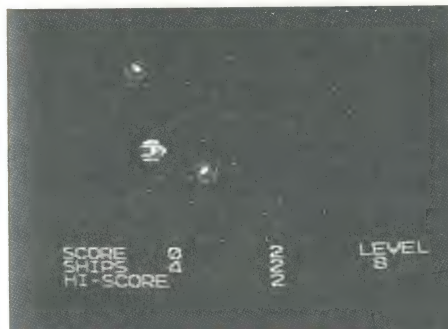
まず、[F４]キーか、[G][A][M][E]　[3][R E T U R N]を押してプログラムを呼びだし（１人は[N]キー、２人は[Y]キーを押す）。ゲームを始めよう。

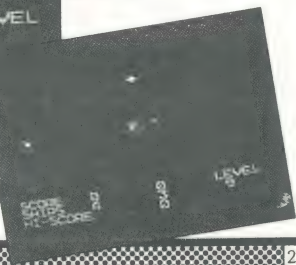
キミが持っている宇宙船は、全部で４隻。そのうちの１隻をコントロールして、ミサイルで敵の宇宙船(10点)、ファイターフライ（５点)、飛行体（１点）をやっつけようぜ！

◀コントローラーの操作によって自分の宇宙船のスピードを変えることも可能だよ

■コントローラーの使い方

ゲーム開始時には✚ボタンを押そう。宇宙船が動き出すぞ。ミサイルは[A]ボタンで発射。[B]ボタンを押すと、ブレーキがかかって宇宙船のスピードが遅くなるけど、そのかわり、こまわりがきくようになるぞ。

■プレイしよう!!

1人用ゲームのときは、自分の宇宙船がなくなった時点でゲームオーバー。2人用のときは、1人の宇宙船がなくなっても、コンピュータがもう1人の宇宙船の相手をしてくれる。

◀相手の宇宙船を撃破しよう。ゲームがひとめぐりすると、自分の宇宙船が1隻増えるぞ。高得点をめざせ!

▶ついに全滅。自分の宇宙船が負けてゲームオーバーだ。もっと腕をきたえよう!

改造プログラムに挑戦しよう!!

手持ちの宇宙船の数を変えてみよう!

1人用ゲームでも、2人用ゲームでも、ゲーム開始時に1人のプレーヤーが持っている宇宙船の数は4隻だ。この数を変えるには、下のリストのSが指定する数を変えればいいんだ。S(0)はプレーヤー1用、S(1)はプレーヤー2用だよ。

腕が上達してきたら、宇宙船の数は少なめにしたほうが面白いだろう。

```
40 L=9:MA=2:N=3:V=7:C=7:S(0)
=4:S(1)=4
```

敵の飛行体の数を変えよう!

画面に現われる飛行体は、最初のままだと同時に2個までしか登場しない。でも、リストのMAの数字を変えれば、その数を1個から6個までに変化させることができるぞ。7以上の数にするとエラーになるから注意。また、0にしても、1個は必ず出てくるぞ。

```
40 L=9:MA=2:N=3:V=7:C=7:S(0)
=4:S(1)=4
```

サンプルプログラム

オリジナルゲームを作ろう!!

……ということ
なんだよ
わかった？

ぜんぜん!!

LOAD
LOADING GAME
OK
チカ
チカ

これは
イン石をよけて
宇宙船が進む
ゲームなんだ

すごーい!!
ほんとに
ゲームだ!!

スタートは
F8
キーのRUN
だよ

はじまりーっ!!
ボォァン!!
このゲーム
おかしいよ!!
すぐ終っちゃう!!
BoSE
おしまい♡
SCORE 0
あっ
あーっ

ほら
おわり♡
だって
STATE!!
ゲームのせいじゃ
ないよ　それは
うっそだぁ
かずよちゃんのうでが…
ちょっと
かしてごらん

できる　でしょ
す
ほんとだー
かして一っ

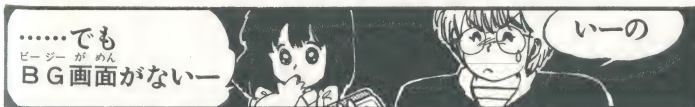
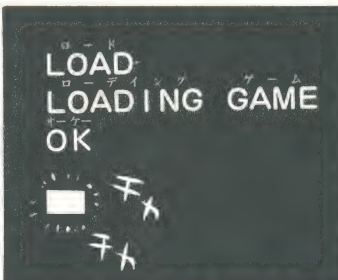
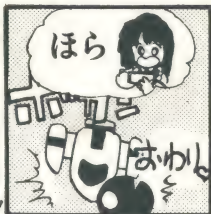
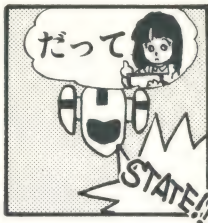
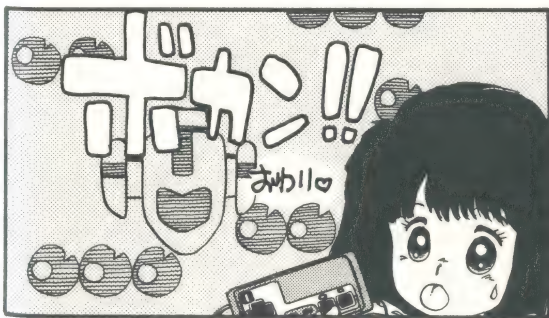
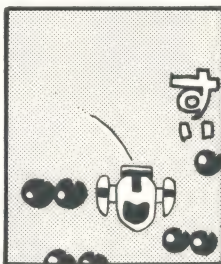
す

ね
す

ボカン
ボカン
ボカン
うぅぅ
ボカン!!
おわり♡

みてみるかい？かずよちゃん
どーやってプログラムしたのぉ？

ねえ このプログラム ほんとうに ベーシックで作ったの？
そーだよ
ふうっ

うんっ!!

こういうふうに F7 のキーを押すと LISTがでてくるんだ
LISTって メモリ中のプログラムを画面に表示することをいうんだよ
FAMILY COMPUTER

```
10 H=0
20 CGSET 1,2
30 CLS
40 DEF SPRITE 0,(2,1,1,0,1)=CHR$
(174)+CHR$(175)+CHR$(172)+CHR$(173
50 X=130
60 SPRITE ON
70 PALETB 0,25,48,33,2
80 A$=CHR$(215)
90 S=0
91 SPRITE 0,X,49
95 PLAY "T2O3M1Y3CRRCRRCRRO4T4C"
100 LOCATE 0,23:PRINT CHR$(226);
110 LOCATE 26,23:PRINT CHR$(226);
140 SPRITE 0,X,49
150 Z=X/8-2
160 IF SCR$(Z,4)<>" " THEN 500
170 IF SCR$(Z+1,4)<>" " THEN 500
175 IF SCR$(Z+2,4)<>" " THEN 500
180 I=STICK(0)
190 IF I=1 THEN X=X+4
200 IF I=2 THEN X=X-4
310 LOCATE RND(24)+1,23
320 IF S>3000 THEN PRINT A$;
330 IF S>7000 THEN PRINT A$;
340 PRINT A$
480 S=S+10
490 GOTO 100
500 LOCATE 10,9
510 PRINT "GAME OVER"
515 PLAY "O1M1Y0T1AT4C"
520 LOCATE 9,12
530 PRINT "SCORE:" ; S
531 IF H>S THEN 535
532 H=S
533 PLAY "T1M1Y3O4CBCBCBT4C"
535 LOCATE 7,14
536 PRINT "HISCORE:" ; H
540 IF STRIG(0)=1 THEN 20
550 IF STRIG(0)=2 THEN 600
560 GOTO 540
600 LOCATE 12,17
610 PRINT "END..."
990 END
```

```
10 H=0
20 CGSET 1,2
30 CLS
40 DEF SPRITE 0,(2,1,1,0,1)=CHR$
(174)+CHR$(175)+CHR$(172)+CHR$(173
```

まず10行ではハイスコアの変数に0を代入しているんだよ

そして　20行で色のセットをして30行で画面をクリアして

40行で宇宙船のスプライトのキャラタクをセットしているんだ

```
50 X=130
60 SPRITE ON
70 PALETB 0,25,48,33,2
80 A$=CHR$(215)
90 S=0
```

50行では
宇宙船の出る場所を
きめていて
60行で
スプライトが使えるように
してある

へー

バックの色は
パレットで
変えていて
Ａ＄にインセキの
キャラクタを
セットしたんだ

91行では　始めに出る宇宙船を出力
していてＰＬＡＹで　スタートの時の
音を出すようになっている
100、110行で
横のカードを出しているんだよ

```
91 SPRITE 0,X,49
95 PLAY "T2O3M1Y3CRRCRRCRRO4T4C"
100 LOCATE 0,23:PRINT CHR$(226);
110 LOCATE 26,23:PRINT CHR$(226);
```

```
140 SPRITE 0,X,49
150 Z=X/8-2
160 IF SCR$(Z,4)<>" " THEN 500
170 IF SCR$(Z+1,4)<>" " THEN 500
175 IF SCR$(Z+2,4)<>" " THEN 500
```

140行で
宇宙船を出して
150行目、ここで
スプライトの座標を
キャラクタの座標と
したんだ

スプライトの座標は
このキャラクタの
座標の8倍あるんだよ

つまり、座標
どうしを合わせるには
8で割ってやれば
いいわけ

いろいろある

うん

とと…

きいてる?

SCR$で変換した
座標をインセキの
キャラクタと判別
しているんだよ

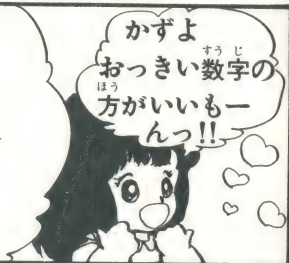
```
180 I=STICK(0)
190 IF I=1 THEN X=X+4
200 IF I=2 THEN X=X-4
```

IF文でIかAか2かで、左右をきめ右だったらXにマイナス4左だったらプラス4にするんだ

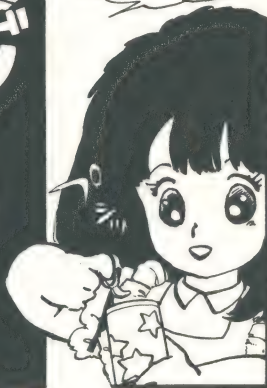
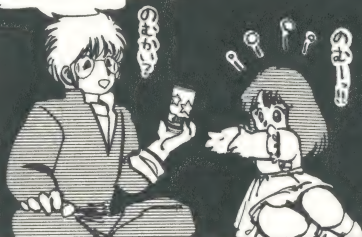
うん

わかる?

```
310 LOCATE RND(24)+1,23
320 IF S>3000 THEN PRINT A$;
330 IF S>7000 THEN PRINT A$;
340 PRINT A$
```

なぜかというと
3000の所と7000の所の
数字を大きくすると
ゲームが簡単になりすぎちゃうし
逆に数字を少なくすると
ムズかしくなりすぎちゃうから
なんだよ

かずよ
おっきい数字の
方がいいもーんっ!!

```
480 S=S+10
490 GOTO 100
```

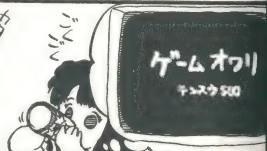
```
500 LOCATE 10,9
510 PRINT "GAME OVER"
515 PLAY "O1M1Y0T1AT4C"
520 LOCATE 9,12
530 PRINT "SCORE:" ; S
```

GAME OVER

SCORE 500

ほら！
かずよちゃん
のゲームの
スコア
が画面に
表示されたよ

このプリントの文字は
好きな文字に
変えることも
できるんだよ

カタカナとかローマ字とかね

ごくごく

531行で
ハイスコアの判定を行なっている
スコアが大きい時は
ファンファーレをならして
Hの変数に今回のスコアを
代入するようにしてあるんだ

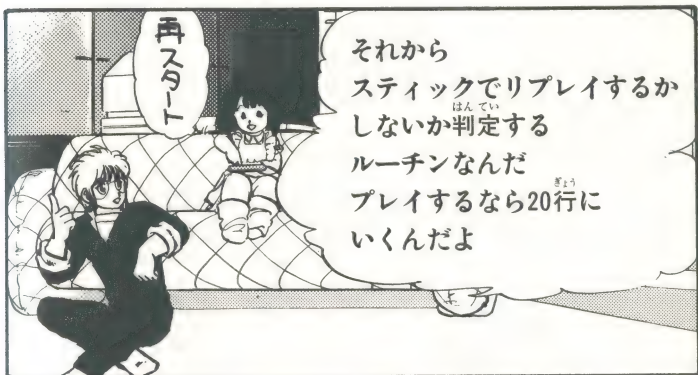
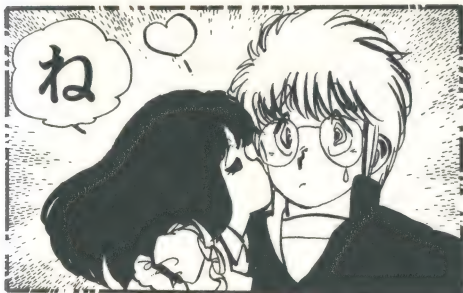
```
531 IF H>S THEN 535
532 H=S
533 PLAY "T1M1Y304CBCBCBT4C"
535 LOCATE 7,14
536 PRINT "HISCORE:" ; H
```

```
540 IF STRIG(0)=1 THEN 20
550 IF STRIG(0)=2 THEN 600
560 GOTO 540
600 LOCATE 12,17
610 PRINT "END..."
990 END
```

わたしも自分のオリジナルゲームを作っちゃうんだあっ!!

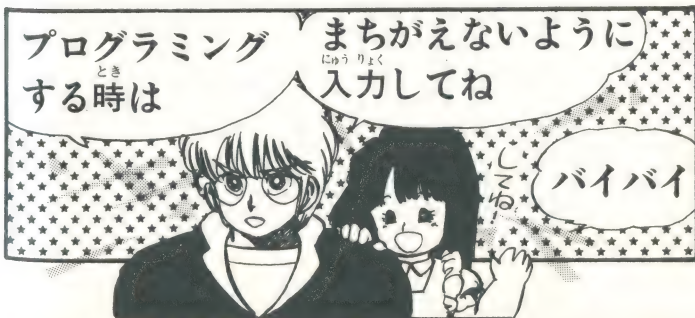
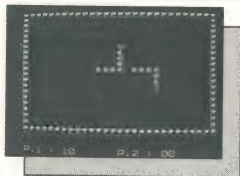
自作!!

すごいでしょ!!

ね

かずよちゃん
オリジナル
ゲームは他にも
まだ
あるんだよ
ほんと

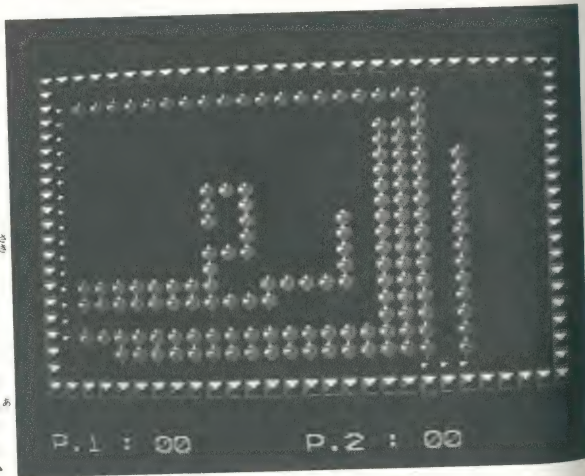
それは、この本に
載ってるのさ
4つも……ね!!
すっごーい♡

ね、どこ
それどこに
あるの!?

プログラミング
する時は
まちがえないように
入力してね
バイバイ
してね!

2 リンゴ陣取りゲーム

2人で遊ぶコンピュータ版〝陣取りゲーム〟だ。相手のリンゴを先に並べられなくした方が勝ちだぞ。さっそくやってみよう！

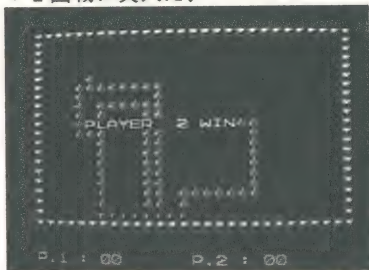
遊び方

まず画面に2つのリンゴが表示されるぞ。片方がキミのリンゴでもう片方は敵のリンゴだ。このリンゴを✚ボタンを押してどんどん並べていこう。ただし、四方のカベに突き当たったら負けだ。並べ方を考えて、敵の動ける範囲をせばめよう。

▶どちらのリンゴが先にカベにぶち当たるかな!?相手の動きを封じ込めよう！

リプレイしよう

リプレイしたいときはスタートボタンを押せばいいぞ。

最初にゲームを始めるときには、RUNでスタート！

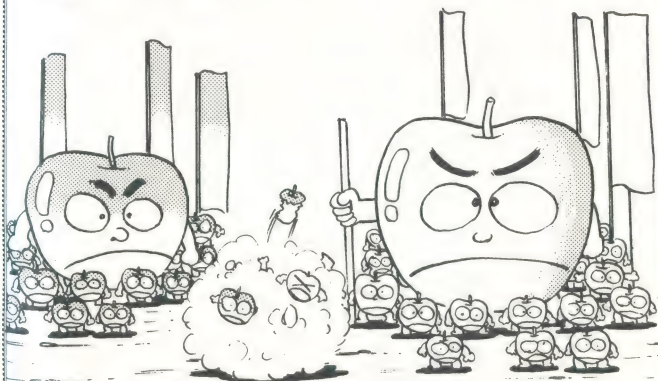
▼2回戦に突入だ！

BGグラフィック

このゲームでは、モード3（配色番号）、Fグループの0番（キャラクタテーブルB）を使ってカベを描いている。キミの好みでカベのもようを変えてみるのもいいぞ。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0	F03	F03	F03	F03	F03	F03	F03	F03	F03	F03	F03	F03	F03	F03	F03	F03	F03	F03	F03	F03	F03	F03	F03	F03	F03	F03	F03	F03
1	F03																											F03
2	F03																											F03
3	F03																											F03
4	F03																											F03
5	F03																											F03
6	F03																											F03
7	F03																											F03
8	F03																											F03
9	F03																											F03
10	F03																											F03
11	F03																											F03
12	F03																											F03
13	F03																											F03
14	F03																											F03
15	F03																											F03
16	F03																											F03
17	F03																											F03
18	F03																											F03
19	F03																											F03
20	F03	F03	F03	F03	F03	F03	F03	F03	F03	F03	F03	F03	F03	F03	F03	F03	F03	F03	F03	F03	F03	F03	F03	F03	F03	F03	F03	F03

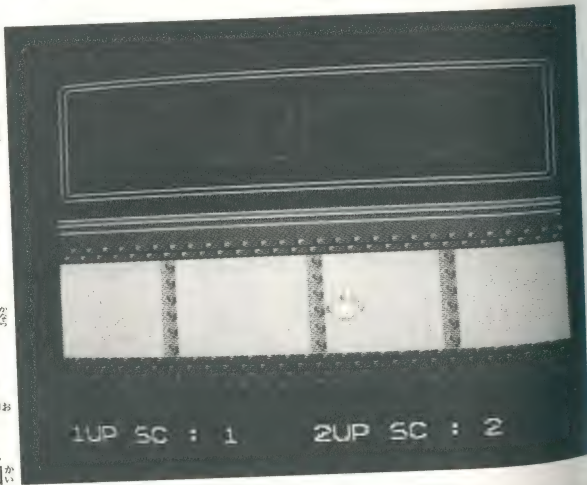
プログラムリスト

```
10 DIM Z(15),X(15),Y(15)
20 Z(1)=1:X(1)=+1
30 Z(2)=1:X(2)=-1
40 Z(4)=1:Y(4)=+1
50 Z(8)=1:Y(8)=-1
60 DIM A(1),B(1),C$(1),D(1)
65 D(0)=2:D(1)=1
70 C$(0)=CHR$(215)
80 C$(1)=CHR$(215)
90 DIM S(1):S(0)=0:S(1)=0
95 PLAY "T1O5M1Y0"
100 CGSET 0,0
110 PALETB 0, 9,37,54,5
200 CLS
210 SPRITE ON
220 VIEW
230 A(0)=10:B(0)=10
240 A(1)=15:B(1)=10
250 LOCATE 0,23
260 PRINT "P.1 :" S(0) "0";
270 LOCATE 14,23
280 PRINT "P.2 :" S(1) "0";
290 K=0
300 FOR I=0 TO 1
310 J=STICK(I)
320 IF Z(J)=0 THEN 350
330 D(I)=J
```

```
350 A(I)=A(I)+X(D(I))
360 B(I)=B(I)+Y(D(I))
370 IF SCR$(A(I),B(I))<>" " THEN
    K=I+1
380 LOCATE A(I),B(I)
390 PRINT C$(I);
400 NEXT
410 IF K>0 THEN 600
490 PLAY "C1"
500 GOTO 300
600 LOCATE 5,10
610 PRINT "PLAYER "K XOR3" WIN"
611 PLAY "T8O2M1Y3C"
615 K=(K XOR3)-1
620 S(K)=S(K)+1
630 IF STRIG(0)<>1 THEN 630
690 GOTO 95
```

3 ニタニタプッシュ

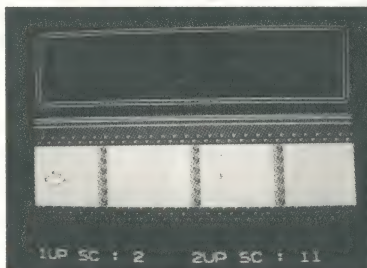
これも2人で遊ぶゲームだ。A、Bボタンを押して、ニタニタを相手の陣地に押し込めよう。ボタンを押すタイミングがむずかしいぞ。

遊び方　ゲームが始まると、画面中央にニタニタが現われるぞ。コルトローラーのA・Bボタンを交互に押すとニタニタが左右に動き始める。プレイヤー1はニタニタを画面左側に、プレイヤー2はニタニタを画面右側に移動させよう。ニタニタを相手陣地に押しやれば勝ちだ。

▶乱数が入っているので、必ずしも押した回数だけニタニタが動くとはかぎらないぞ

リ プレイしよう

もう1度ゲームをくり返したいときは、スタートボタンを押そう。

画面の下にはそれぞれの勝った回数が表示されるぞ。

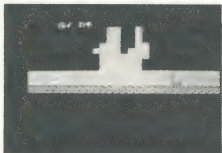
▼ニタニタを相手陣地に押しやろう

Gグラフィック

空はFグループの6番、配色番号0、柱はHグループの5番、配色番号0、下のレンガはFグループの1番、配色番号3番、上のレンガはFグループの3番、配色番号3番を使用。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60
1	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60
2	M60	M60	I60	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	I70	M60	M60
3	M60	M60	J20		M	A	R	I	O	'	S		:	0	0	M		R	U	N	N	I	N	G		J20	M60	M60
4	M60	M60	J00	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J30	J10	M60	M60
5	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60
6	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60
7	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60
8	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60
9	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60
10	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60
11	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60
12	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60
13	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60
14	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60
15	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60
16	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60	M60
17	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31
18	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31
19	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31
20	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31

プログラムリスト

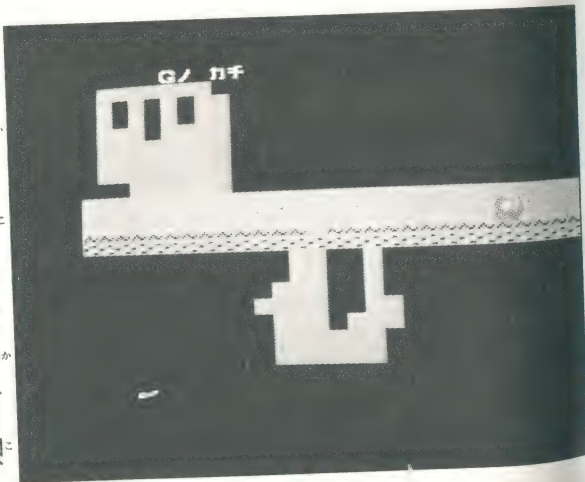
```
10 DIM S(2)
20 VIEW
100 DEFMOVE(0)=SPRITE(11,0,1,1,0
    )
110 POSITION 0,120,144
120 MOVE 0
130 SPRITE ON
135 CLS
136 VIEW
140 CGSET 1,2
150 J=4
160 LOCATE 0,23:PRINT "1UP SC :"
    S(2);
170 LOCATE 14,23:PRINT "2UP SC :
    " S(1);
180 PLAY "T2O3M1Y0C1E4D2GR1C1R1A
    4"
190 PLAY "T1O3M1Y2R1"
200 I=STRIG(0)
201 IF G2=I THEN 220
202 G2=I
203 IF MOVE(0)=-1 THEN 220
210 IF I<>J THEN 220
211 DEFMOVE(0)=SPRITE(11,7,1,8,0
    )
212 MOVE 0
214 J=12XORJ
```

```
215 PLAY "E
220 I=STRIG(1)
221 IF G1=I THEN 240
222 G1=I
223 IF MOVE(0)=-1 THEN 240
230 IF I<>J THEN 240
231 DEFMOVE(0)=SPRITE(11,3,1,8,0
    )
232 MOVE 0
234 J=12XORJ
235 PLAY "G
240 IF XPOS(0)>208 THEN O=1:PLAY
     "O2T3GAGAGAGAG4":GOTO 300
250 IF XPOS(0)<32  THEN O=2:PLAY
     "O2T3EBEBERBRE4":GOTO 300
260 GOTO 200
300 S(O)=S(O)+1
310 LOCATE 0,23:PRINT "1UP SC :"
     S(2);
320 LOCATE 14,23:PRINT "2UP SC :
    " S(1);
390 IF STRIG(0)<>1 THEN 390
400 GOTO 100
```

4 ジャンケンポン

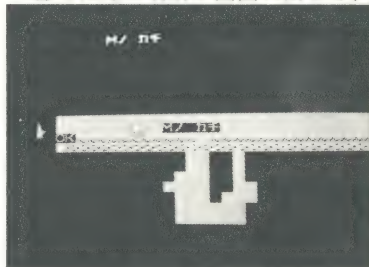
コンピュータでやるジャンケンゲームだ。✚ボタンの上か下を押すとチョキ、右を押すとパー、左を押すとグーが出るぞ。2人で遊ぼう！

遊び方

ジャン、ケン、ポンと画面に表示されたら、✚ボタンを押してグーチョキパーのいずれかを出そう。プレイヤー1が勝つとペンペンが、プレイヤー2が勝つとファイアーボールが、画面の右から左へ進んでいくぞ。自分のキャラクタが先に左端へ着けば勝ちだ。

グーで勝てば1個、チョキで勝てば2個、パーで勝てば5個、自分のキャラが進むぞ。

リ プレイしよう

どちらかのキャラクタが左端に着いたらゲームセット。もう1度やりたいときはRUNを入力しよう。こんどはどっちが勝つかな？

▼勝った方の名前が画面に出るぞ！

B Gグラフィック

グーチョキパーの色は上段のプレイヤー1用が配色番号2を、下段のプレイヤー2用が配色番号3を使用している。ほかの色と組み合わせてもいいぞ。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0																												
1		M52	M52	M52	M52	M52	M52	M52					M62	M62			M62						M72	M72				
2		M52		M52		M52		M52	M52				M62	M62			M62					M72	M72	M72	M72			
3		M52		M52		M52		M52	M52				M62	M62			M62					M72	M72	M72	M72			
4		M52	M52	M52		M52	M52	M52	M52			M62	M62	M62			M62	M62			M72	M72	M72	M72	M72		M72	
5		M52	M52	M52	M52	M52	M52	M52	M52		M62	M62	M62	M62		M62	M62	M62			M72	M72	M72	M72	M72	M72	M72	
6		M52	M52	M52	M52	M52	M52	M52	M52			M62	M62	M62	M62	M62	M62	M62			M72	M72	M72	M72	M72	M72		
7		M52	M52	M52	M52	M52	M52	M52	M52			M62	M62	M62	M62	M62	M62				M72	M72	M72	M72	M72	M72		
8				M52	M52	M52	M52	M52	M52			M62	M62	M62	M62	M62	M62					M72	M72	M72	M72			
9	M53	M53	M53	M53	M53	M53	M53	M53	M53	M53	M53	M53	M53	M53	M53	M53	M53	M53	M53	M53	M53	M53	M53	M53	M53	M53	M53	M53
10	M53	M53	M53	M53	M53	M53	M53	M53	M53	M53	M53	M53	M53	M53	M53	M53	M53	M53	M53	M53	M53	M53	M53	M53	M53	M53	M53	M53
11	I13	I13	I13	I13	I13	I13	I13	I13	I13	I13	I13	I13	I13	I13	I13	I13	I13	I13	I13	I13	I13	I13	I13	I13	I13	I13	I13	I13
12	G43	G43	G43	G43	G43	G43	G43	G43	G43	G43	G43	G43	G43	G43	G43	G43	G43	G43	G43	G43	G43	G43	G43	G43	G43	G43	G43	G43
13			M51	M51	M51	M51	M51	M51					M61	M61			M61						M71	M71				
14		M51		M51		M51		M51	M51				M61	M61			M61					M71	M71	M71	M71			
15		M51		M51		M51		M51	M51			M61	M61	M61			M61					M71	M71	M71	M71			
16		M51	M51	M51		M51	M51	M51	M51		M61	M61	M61	M61			M61	M61			M71	M71	M71	M71	M71		M71	
17		M51	M51	M51	M51	M51	M51	M51	M51			M61	M61	M61			M61	M61			M71	M71	M71	M71	M71	M71	M71	
18		M51	M51	M51	M51	M51	M51	M51	M51			M61	M61	M61		M61	M61				M71	M71	M71	M71	M71	M71		
19		M51	M51	M51	M51	M51	M51	M51	M51			M61	M61	M61	M61	M61	M61				M71	M71	M71	M71	M71	M71		
20				M51	M51	M51	M51	M51	M51													M71	M71	M71	M71			

プログラムリスト

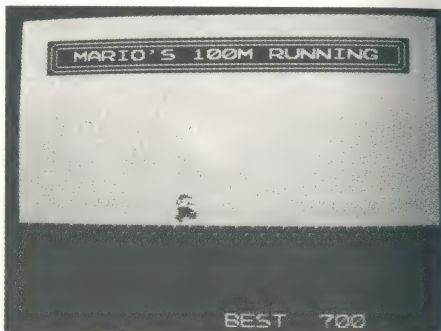
```
10 CLS
20 SPRITE ON:CGSET 1,1
30 DIM  X(1),C(1),H(1),PA(1)
40 X(0)=224:X(1)=224
50 N2$="コンピュータ"
60 GOSUB 340:CLS: VIEW
70 PALETB 1,13,13,13,13
80 PALETB 2,13,13,13,13
90 GOSUB 570
100 H(0)=0:H(1)=0
110 FOR N=0 TO 1
120 S=STICK(N)
125 IF A$="Y" AND N=1 THEN S=RND(4)+1
130 IF S=1 THEN PA(N)=3:C(N)=5:H(N)=1
140 IF S=2 THEN PA(N)=1:C(N)=1:H(N)=1
150 IF S=4 OR S=8 THEN PA(N)=2:C(N)=2:H(N)=1
160 NEXT
170 IF H(0)+H(1)<>2 THEN 110
180 FOR N=0 TO 1
190 GOSUB 290
200 NEXT
210 IF PA(0)=PA(1) THEN PAUSE 40:GOTO 70
220 IF PA(0)=1 AND PA(1)=2 THEN 390
230 IF PA(0)=1 AND PA(1)=3 THEN 440
240 IF PA(0)=2 AND PA(1)=1 THEN 440
250 IF PA(0)=2 AND PA(1)=3 THEN 390
260 IF PA(0)=3 AND PA(1)=1 THEN 390
270 IF PA(0)=3 AND PA(1)=2 THEN 440
280 GOTO 70
290 ON PA(N) GOTO 300,310,320
300 PALETB N+1,13,1,13,13:GOTO 330
310 PALETB N+1,13,13,5,13:GOTO 330
320 PALETB N+1,13,13,13,9:GOTO 330
330 RETURN
340 INPUT "コンピュータ ト シマスカ?",A$
350 IF A$="Y" THEN INPUT"キミノ ナマエハ?",N1$:GOTO
    380
```

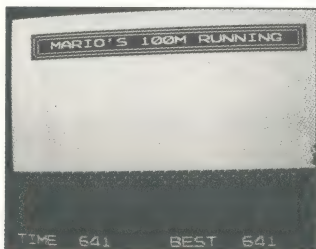
```
360 INPUT"プレイヤ-1 ノ ナマエハ?",N1$
370 INPUT"プレイヤ-2 ノ ナマエハ?",N2$
380 RETURN
390 LOCATE 5,0
400 PRINT N1$;"ノ カチ         "
410 N=0
420 GOSUB 490
430 GOTO 70
440 LOCATE 5,0
450 PRINT N2$;"ノ カチ         "
460 N=1
470 GOSUB 490
480 GOTO 70
490 DEF MOVE(N)=SPRITE(N+4,7,5,C(N)*4,0,0)
500 POSITION N,X(N),100
510 MOVE N
520 PLAY"BA"
530 IF MOVE(N)=-1 THEN 520
540 X(N)=X(N)-C(N)*8
550 IF X(N)<1 THEN 710
560 GOTO 70
570 LOCATE 5 ,0
580 PRINT "                "
590 LOCATE 13,0
600 PRINT "ジャン"
610 PLAY"CD
620 PAUSE 10
630 LOCATE 13,0
640 PRINT "ケン "
650 PLAY"EC
660 PAUSE 10
670 LOCATE 13,0
680 PRINT "ポン "
690 PLAY"D
700 RETURN
710 LOCATE 10,10
720 IF N=0 THEN PRINT N1$;"ノ カチ"
730 IF N=1 THEN PRINT N2$;"ノ カチ"
740 END
```

5 マリオの100mランニング

コントローラーのA・Bボタンを交互に押してマリオを走らせよう。キミはどれくらいのタイムでマリオを走らせることができるかな？

び方

画面の左側にいるマリオを右側に向かって走らせよう。A・Bボタンを交互に押すとマリオは走り出すぞ。最高タイムで走らせればファンファーレが鳴るよ。

キミのマリオは100mを何秒で走破できるかな。ベストタイムをめざそう！

リプレイしよう

スタートボタンを押すと、マリオは最初の位置に戻るぞ。画面下に表示されているのがいままでのベストタイムだ。新記録に挑戦しよう。

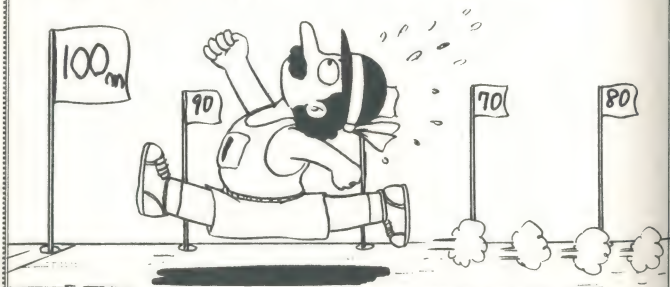
Gグラフィック

レンガは配色番号3、Fグループの3番を使っているんだ。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0	I63	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	I73
1	J23																											J23
2	J23		K73	K53	L03	K73	K53	L03	L03		K73	K73	K53	L03	L03		K53				K73	K53	L03	K73	K53	L03		J23
3	J23		K63		K63	K63		K63	K63		K63	K63		K63	K43	L23					K63		K63	K63		K63		J23
4	J23		K43	K53	L23	K63		K63	K63	K63	K63	K43	K53	L23	K63				K53		K63		K63	K63		K63		J23
5	J23		K63			K63		K63	K63	K63	K63	K63			K63						K63		K63	K63		K63		J23
6	J23		K63			L13	K53	L23	L13	K13	L23	L13	K53	L23	K63						L13	K53	L23	K63		K63		J23
7	J23																											J23
8	J03	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J33	J13
9																												
10	F53	F53	F53	F53	F53	F53	F53	F53	F53	F53	F53	F53	F53	F53	F53	F53	F53	F53	F53	F53	F53	F53	F53	F53	F53	F53	F53	F53
11	M73	M73	M73	M73	M73	M73	M73	M73	M73	M73	M73	M73	M73	M73	M73	M73	M73	M73	M73	M73	M73	M73	M73	M73	M73	M73	M73	M73
12	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33	F33
13	M60	M60	M60	M60	M60	M60	H50	M60	M60	M60	M60	M60	M60	M60	H50	M60	M60	M60	M60	M60	M60	H50	M60	M60	M60	M60	M60	M60
14	M60	M60	M60	M60	M60	M60	H50	M60	M60	M60	M60	M60	M60	M60	H50	M60	M60	M60	M60	M60	M60	H50	M60	M60	M60	M60	M60	M60
15	M60	M60	M60	M60	M60	M60	H50	M60	M60	M60	M60	M60	M60	M60	H50	M60	M60	M60	M60	M60	M60	H50	M60	M60	M60	M60	M60	M60
16	M60	M60	M60	M60	M60	M60	H50	M60	M60	M60	M60	M60	M60	M60	H50	M60	M60	M60	M60	M60	M60	H50	M60	M60	M60	M60	M60	M60
17	M60	M60	M60	M60	M60	M60	H50	M60	M60	M60	M60	M60	M60	M60	H50	M60	M60	M60	M60	M60	M60	H50	M60	M60	M60	M60	M60	M60
18	M60	M60	M60	M60	M60	M60	H50	M60	M60	M60	M60	M60	M60	M60	H50	M60	M60	M60	M60	M60	M60	H50	M60	M60	M60	M60	M60	M60
19	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31
20	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31	F31

プログラムリスト

```
30 CGSET 1,1
40 L=700
100 CLS
105 K=0
110 SPRITE ON
120 VIEW
140 DEFMOVE(0)=SPRITE(0,3,1,2,0)
150 POSITION 0,16,141
```

```
151 MOVE 0
155 I=4
156 LOCATE 14,23:PRINT "BEST " L
    ;
158 PLAY "T3O3M1Y0C4RCRCRT1O4C1"
159 PLAY "T1O2M1Y1R1"
160 MOVE 0
170 K=K+1:IF STRIG(0)<>I THEN 17
    0
180 I=I XOR12
185 IF XPOS(0)>218 THEN 200
186 PLAY "A"
190 GOTO 160
200 LOCATE 0,23:PRINT "TIME " K;
205 IF L>K THEN L=K:PLAY "T3O5CB
    CBCRBRT5M1Y1C"
210 LOCATE 14,23:PRINT "BEST " L
    ;
220 IF STRIG(0)<>1 THEN 220
230 GOTO 100
```

ファミリーベーシック コマンド早見表

```
LIST
10 A=10
20 B=-20
30 GOSUB 100
40 PRINT "*** SGN
50 A=SGN(A)
60 B=SGN(B)
70 GOSUB 100
80 END
100 PRINT "A=";A
110 PRINT "B=";B
120 RETURN
OK
RUN
A=
B=-20
*** SGN ***
A= 1
B=-1
OK
```

キミの必要としている命令がすぐに探し出せるぞ！

●CLEAR〔CLE.〕

通常プログラムの先頭部分に書かれ、サブルーチンの中では、使用できない。また、一度指定したアドレスはふたたびCLEARで指定されるまで有効である。 →99ページ

●NEW

メモリに入っている古いプログラムを消す働きをする。同時に変数の内容もすべて消してしまうので注意。省略形はない。プログラム上では使用しないこと。 →67、149ページ

●LIST〔L.〕

メモリに入っているプログラムを画面上に出力させるときに使う。ESCキーを押すと、表示を一時停止させることができる。 →66ページ

●RUN〔R.〕

プログラムを実行させるためのコマンド。ただし、変数はすべてクリアされる。プログラムを途中から実行したいときは、実行開始行番号をつける。 →66ページ

●CONT〔C.〕

停止しているプログラムを再開するときに使用。止まった次の行番号の命令文からプログラムは実行を開始する。

→67ページ

●LOAD〔LO.〕

カセットテープに記録されているプログラムをメモリに記録する。ファイル名の指定にしたがってプログラムを読み込む働きをする。 →62ページ

●SAVE〔SA.〕

メモリに記録されたプログラムをカセットテープに記録す

る。LOADと逆の働きをする。 →103ページ

●LOAD ?〔LO.? またはLO.P.〕

SAVEしたプログラムがファイルに正しく記録されたかどうかを確認するときに使う。正しく記録されていればOKが表示され、記録されていなければTPエラーとなる。

●PRINT〔?またはP.〕

画面に定数、数値変数の値、文字変数の値(文字列)、式の値(演算結果)を表示する。 →58、120、130ページ

●INPUT〔I.〕

キーボードから入力した数値や文字のデータを変数に入れる。「,」を入力する場合は「"」で囲むこと。 →131ページ

●LINPUT〔LIN.〕

INPUTと同様、キーボードから文字を入力する。ただし、入力できるのは31文字以内の文字列だけ。「,」を「"」で囲む必要はない。 →58、131ページ

●DIM〔DI.〕

配列の名前と次元数、それに添字のサイズを設定する。1つのDIM文で複数の配列を宣言でき、配列ごとにメモリの範囲内で2次元までの添字指定が可能。配列宣言後、すべての配列の中身はクリアされる。 →137ページ

●GOTO〔G.〕

プログラムを指定した行番号へ無条件にジャンプさせる。ダイレクト命令では、定数の値をクリアせずに、特定行番号からプログラムを実行させることが可能。 →59、132ページ

●GOSUB〔GOS.〕

プログラム内のサブルーチンを呼び出す。呼び出されるサ

ブルーチンの最後には必ずＲＥＴＵＲＮをつけること。

→59、134ページ

●ＲＥＴＵＲＮ〔ＲＥ.〕

　ＧＯＳＵＢで呼ばれたサブルーチンを復帰させる。

→59ページ

●ＩＦ－ＴＨＥＮ〔ＩＦ－Ｔ.〕

　論理式による条件分岐をおこなう。ＩＦ文の中に別のＩＦ文を入れてもかまわない。→60、133ページ

●ＦＯＲ～ＴＯ～ＳＴＥＰ／ＮＥＸＴ〔Ｆ.-ＴＯ. ＳＴ.／Ｎ〕

　ＦＯＲ～ＮＥＸＴループの間の処理をくり返しおこなう。ＮＥＸＴの後にはループ変数名をつけることはできない。

→61、135ページ

●ＯＮ〔Ｏ.〕

　式の値によって指定された行へジャンプする。式の値が０のときや指定した行番号の個数を越えるときは、ＯＮ文の次の文に移る。→63ページ

●ＳＴＯＰ〔ＳＴＯ.〕

　プログラムの実行を止める。このとき、変数の中はクリアされない。→64ページ

●ＥＮＤ〔Ｅ.〕

　プログラムの実行を終了させる。最終行のＥＮＤは省略できる。→64ページ

●ＳＷＡＰ〔ＳＷ.〕

　２つの変数の内容を交換するときに使う。ただし、交換する変数の型は、一致していなければならない。→64ページ

●ＲＥＭ〔’〕

プログラム文にコメントを入れる。REM文で書かれたコメントは、プログラムの実行には関係してこない。コメントをそう入するためのステートメントである。

●READ〔REA.〕

DATA文で用意されたデータをREAD文の変数に入力する。 →62ページ

●DATA〔D.〕

READで読み込むデータを用意する。READとは必ず対になっていなければならない。 →62、138ページ

●RESTORE〔RES.〕

READ文で読むDATA文を指定する。これを使えば、同じDATA文を何度も利用したり、読み込むDATA文を好きなように指定することができる。

●CALL〔CA.〕

マシン語サブルーチンを呼び出す。マシン語サブルーチンは、CLEAR文で指定したアドレス以降に置くこと。

→99ページ

●POKE〔PO.〕

メモリ上の指定したアドレスに1バイトのデータを書き込む。 →98ページ

●LOCATE〔LOC.〕

バックグラウンド面のカーソル位置を指定するコマンド。

→74、130ページ

●COLOR〔COL.〕

バックグラウンド面に表示する文字の配色番号を画面上のエリア毎に指定する。 →76ページ

●CGEN〔CGE.〕

バックグラウンド面とスプライト面にキャラクタの割り当てを決める。 →75ページ

●CLS〔CL.〕

バックグラウンド面をクリアする。 →75ページ

●CGSET〔C G.〕

ＢＧやスプライトで使用するパレットの割り当てを決める。 →77、146ページ

●PALET〔PAL.B, PAL.S.〕

背景の色やキャラクタの表示色を色コードの中から選んで着色する。ＣＧＳＥＴ文で決められたパレットに対して実行される。 →78、146、172ページ

●DEF MOVE〔DE.M.〕

アニメキャラクタに対して固有の動きを定義する。 →48、148ページ

●MOVE〔M.〕

ＤＥＦ ＭＯＶＥで定義されたアニメキャラクタの動きを開始させる。 →50ページ

●CUT〔CU.〕

ＭＯＶＥ文で動きを開始したアニメキャラクタの動きをストップさせる。 →51ページ

●ERA〔ER.〕

ＭＯＶＥ文で動きを開始させたアニメキャラクタをストップさせ、スプライト面上から消す。 →51ページ

●POSITION〔POS.〕

ＭＯＶＥ文でアニメキャラクタを動かす前に、動き始めの

初期座標を与える。 →50ページ

●XPOS〔XP.〕

DEF　MOVEで定義された動作番号の位置のヨコ方向の座標値を求める。 →53ページ

●YPOS〔YP〕

DEF　MOVEで定義された動作番号の位置のタテ方向の座標値を求める。 →53ページ

●MOVE(n)〔M.(n)〕

MOVE文で動きを開始させた動作番号のアニメキャラクタが、DEF　MOVEで定義した動きを完了したかどうかを、この関数に値で与える。 →50ページ

●KEY〔K.〕

ファンクションキーに対して文字列を定義する。 →177ページ

●KEYLIST〔K.L.〕

ファンクションキーの定義状態を画面に表示する。 →177ページ

●SYSTEM〔S.〕

ベーシック実行モードをゲームベーシックモード画面に戻す。 →162ページ

●VIEW〔V.〕

BGグラフィック面を、バックグラウンド面にコピーする。 →74ページ

●BEEP〔B.〕

「ピー」という音を出す。 →83、123ページ

●PLAY〔PL.〕

音楽を演奏する。 →81、190、194ページ

●ABS〔AB.〕

数式の絶対値を与える。 →86ページ

●SGN〔SG.〕

数式の符号を与える。 →86ページ

●RND〔RN.〕

引数未満の乱数を発生する。 →86、136ページ

●ASC〔AS.〕

文字コードを数値に変換する。 →87ページ

●CHR$〔CH.〕

数値を文字に変換する。 →87、209ページ

●VAL〔VA.〕

数字の文字列を数値に変換する。 →88ページ

●STR$〔STR.〕

数式の値を文字に変換する。

●HEX$〔H.〕

数式を16進数の文字列に変換する。 →88ページ

●LEFT$〔LEF.〕

文字列の左側から、指定した数だけ文字を取り出す。

→89ページ

●RIGHT$〔RI.〕

文字列の右側から、指定した数だけ文字を取り出す。

→89ページ

●MID$〔MI.〕

文字列の中から、指定した数だけ一連の文字を取り出す。

→89ページ

●LEN〔LE.〕

文字列の文字数を与える。 →90ページ

●PEEK〔PE.〕

指定したメモリアドレスからデータを取り出す。

→98ページ

●POS

画面のカーソルの水平位置を与える。省略形はない。

→50ページ

●FRE〔FR.〕

ベーシックのプログラムでまだ使用していないユーザーメモリのサイズを与える。 →68、178ページ

●STICK〔STI.〕

コントローラーの✚ボタンの方向を示す。→94、131ページ

●STRIG〔STR.〕

コントローラーのトリガーボタンの情報を示す。

→95ページ

●CSRLIN〔CSR.〕

カーソルの垂直位置を与える。 →92ページ

●SCR$〔SC.〕

BGグラフィック画面で表示されている文字、絵の座標と配色番号を求める。 →93、210ページ

●INKEY$〔INK.〕

キーボードから入力された1文字を与える。→131、132ページ

●DEF SPRITE〔DE.SP.〕

スプライト面上に表示したいアニメキャラクタを定義する。

→49、144ページ

●SPRITE〔SP.〕

定義されたスプライトを好きな場所に表示、あるいは消去する。 →52、144ページ

●SPRITE ON〔SP.O.〕

スプライト表示モード。スプライト面をバックグラウンド面に重ねて表示できるようにする。 →48、144ページ

●SPRITE OFF

スプライト面の表示を取り消す。スプライト面に表示されているすべてのアニメキャラクタが、これによって見えなくなる。

◇V3に追加されたコマンド

●AUTO〔A.〕

入力するプログラムの行番号を自動的に付けてくれるコマンド。 →69ページ

●BACKUP〔BA.〕

プログラムやBG面をバックアップする。→105、106ページ

●BGGET〔BGG.〕

BG面1をバックアップする。 →106ページ

●BGPUT〔BGP.〕

BGGETでバックアップされたBG面をVRAMへ転送する。

●BGTOOL〔BG.〕

ベーシックモードからBGグラフィックモードへ移行する。

→162ページ

●CAN

DEF MOVEで動きを定義したアニメキャラクタを未

定義にする。 →54ページ

●CLICK〔CLI.O.／CLI.OF.〕

クリック音を出したり消したりする。 →68ページ

●CLS〔CL.〕

BG面の表示をクリアする。 →75ページ

●CRASH〔CR.〕

DEF MOVEによるアニメキャラクタの重なりを判定する。 →55ページ

●DELETE

プログラム中の消したい部分を、行番号の指定でまとめて取り消すことができる。 →70ページ

●ERL

エラーの発生した行番号が求められる。 →72ページ

●ERR

エラーが発生したエラーコードが求められる。→72ページ

●ERROR〔ERR.〕

仮のエラーを発生させる。 →72ページ

●FIND〔FI.〕

プログラムの中から、指定した文字列を探し、それをふくむ行を表示する。 →71ページ

●FILTER〔FIL.〕

BG面の全面着色をおこなう。 →79ページ

●GAME〔GA.〕

ROMに内蔵されているゲームが実行される。→222ページ

●INSTR〔INS.〕

文字列1の中で、指定した文字列2を求める。→90ページ

この本に掲載された文章、写真、イラストについての責任は、総て小社にあります。お問合わせ、ご質問は下記編集部までお願いいたします。

任天堂ファミリーベーシック™大百科

発 行 人　加納　将光
編 集 人　長渇　謙彰
発 行 所　株式会社　勁文社
〒164 東京都中野区本町３丁目32番15号
☎372-3281(編集)・372-3291(営業)振替
東京9-13311番
写植・版下　株式会社　パンアート
印 刷 所　株式会社　放光社
製 本 所　明興製本工業株式会社
表紙デザイン　ホワイトアート(田代晴美)

落丁、乱丁本は当社にておとりかえいたします。
発行日は、カバーに表示してあります。

定価650円

ケイブンシャの大百科

1 60年版

全怪獣怪人大百科

史上最多！特撮147番組3823体の怪獣怪人紹介！

定価650円

2 85年版

プロ野球大百科

セ・パ全選手名鑑とプロ野球の全てを大特集。

定価650円

4 '85年版

ヤングタレント大百科

キミだけに教えるタレント㊙情報を満載！

定価650円

6

野球ルール大百科

キミのために実戦ルールをわかりやすく解説。

定価650円

11

大　相　撲大百科

ニューパワー小錦を始め、大相撲のすべて!!

定価650円

24

つり入門大百科

川づり、海づり、ルアーづりの全てを満載。

定価600円

26 ウルトラ兄弟の全て

ウルトラマン大百科

ウルトラQからレオまで全323話を収録！

定価600円

30 ラジコンを知ろう！

ラジコン大百科

作り方から操縦法、作品の数々を紹介。

定価600円

39 ウルトラ怪獣の全て

続・ウルトラマン大百科

ウルトラ怪獣424体を足型つきで大紹介。

定価600円

70 Nゲージの全て

鉄道模型大百科

レイアウトの作り方、車両編成などを解説。

定価650円

小さくて、あつくて、楽しい本

76 1985年版

全アニメ大百科

アニメの熱い歴史!!
694作品を全て紹介!!
定価650円

122

世界の怪獣大百科

実在の怪獣に加えアニメや伝説の怪獣も紹介。
定価650円

100 プラモ工作に挑戦

プラモデル入門大百科

初心者から上級者まで、プラモの決定版。
定価650円

123 恐怖の実体験!

妖怪・幽霊大百科

幽霊の謎、妖怪の秘密を徹底分析。名鑑つき。
定価650円

110

マイコン大百科ゲーム編

最新コンピュータゲームの攻略方法大公開!
定価650円

128

ラジコンテクニック大百科

マニアも楽しめるラジコンの遊び方いろいろ。
定価650円

112

機動戦士ガンダムメカ大百科

ガンダムに登場するメカのすべてを大特集!
定価650円

135

プラモデルテクニック大百科

人気のホンダ・シティからアラレちゃんまで。
定価650円

113 59年版

巨人軍大百科

創立50年!日本一をめざすジャイアンツのすべて。
定価650円

137

サッカー入門大百科

サッカーの実戦テクニックをわかりやすく解説。
定価650円

ケイブンシャの大百科

199

チョロ獣 大百科

チョロ獣、チョロQの遊び方と改造法大特集！

定価650円

200

鉄道模型 Nゲージ・大百科

人気車両の改造法から遊び方、作り方大公開

定価650円

201

まんがイラスト 大百科

プロ顔負けのまんがイラストの描き方を紹介。

定価650円

202

宇宙刑事シャイダー 大百科

シャイダーの必殺技と不思議界フーマの全て。

定価650円

203

日本のミステリーゾーン 大百科

謎が渦まく日本のミステリーポイントを探る。

定価650円

204

動物びっくり超能力 大百科

あっとおどろく世界のビックリ動物を紹介。

定価650円

205

ジャッキー・チェン スーパーアクション大百科

新作スパルタンXとジャッキーアクションの全て。

定価650円

206

日本 謎の伝説 大百科

日本古代より伝来の謎の神話・伝説を大特集。

定価650円

207

最新メカ ラジコンバギー 大百科

走行テクニックとベストセッティング研究！

定価650円

208

ゴジラ 大百科

よみがえった怪獣王ゴジラの全てを大特集！

定価650円

小さくて、あつくて、楽しい本

209 決定版

なぞなぞクイズ大百科

650問の難問に挑戦！
キミはクイズ博士だ!!

定価650円

210 任天堂

ファミリーコンピュータ大百科

最新人気ゲームの必勝法と遊び方を大特集！

定価650円

211 完全版

ガンダムプラモ2大百科

最新モビルスーツ総登場！ガンプラ決定版!!

定価650円

212

スーパーメカ ロボット大百科

最新ロボットのスーパーメカニック大研究!!

定価650円

213 "チャンピオン魔"

ユン・ピョウ大百科

若きスーパースターユン・ピョウの魅力爆発！

定価650円

214

迷路パズル大百科

難問・奇問がいっぱい。
キミの頭脳に挑戦だ！

定価650円

215

日本の妖怪大百科

ろくろくびを始め、日本の妖怪300体を紹介。

定価650円

216 85年版

ラジコンカタログ大百科

陸海空のラジコン作品を集めた決定版！

定価650円

大百科シリーズ・バックナンバー

195 ヒサクニヒコの
おもしろ工作大百科 650円
198 秘境アマゾン大百科 650円

118 カラー版世界の昆虫大百科 650円
120 世界の超特急大百科 650円
129 大戦隊ゴーグルV大百科 650円
130 まんがおもしろクイズ
大百科 650円
131 四季のおりがみ大百科 650円
132 ヒーロー雑学もの知り
大百科 650円
134 宇宙刑事ギャバン大百科 500円
136 アントニオ猪木・
タイガーマスク大百科 500円
139 占いあそび大百科 650円
140 太陽の牙ダグラム大百科 650円
141 ヒーローメカ大百科 650円
144 モデルガン大百科 650円
145 少林寺拳法入門大百科 650円
146 ジャッキー・チェン大百科 650円
149 機動戦士ガンダムプラモ
大百科 650円
150 異星人・UFO大百科 650円
151 ブルース・リー大百科 650円
152 プロレス大百科
PART 3 650円
153 野球実戦テクニック大百科 650円
155 ザブングル・ダグラムメカ
大百科 650円
157 ジャッキー・チェン2大百科 650円
159 科学戦隊ダイナマン大百科 650円
161 世界の超予言大百科 650円
162 からだなぜなに大百科 650円
163 プラモカー・オートバイ
大百科 650円
164 アニメアイドル大百科 650円
165 忍者・忍法大百科 650円
166 クンフースター大百科 650円
169 宇宙刑事シャリバン大百科 650円
171 ジャッキー・チェン3大百科 650円
173 林葉直子の強くなる将棋
大百科 650円
175 プラモ戦車軍団大百科 650円
176 地球の謎大百科 650円
177 人気ヒーロープラモ大百科 650円
180 おもしろ科学大百科 650円
181 ウンチの大百科 650円
183 オリンピック大百科 650円
186 ジャッキーチェンずっこけ
クンフー大百科 650円
189 昆虫もの知り大百科 650円
190 決定版プロレス大百科 650円
191 おもしろ日本一大百科 650円

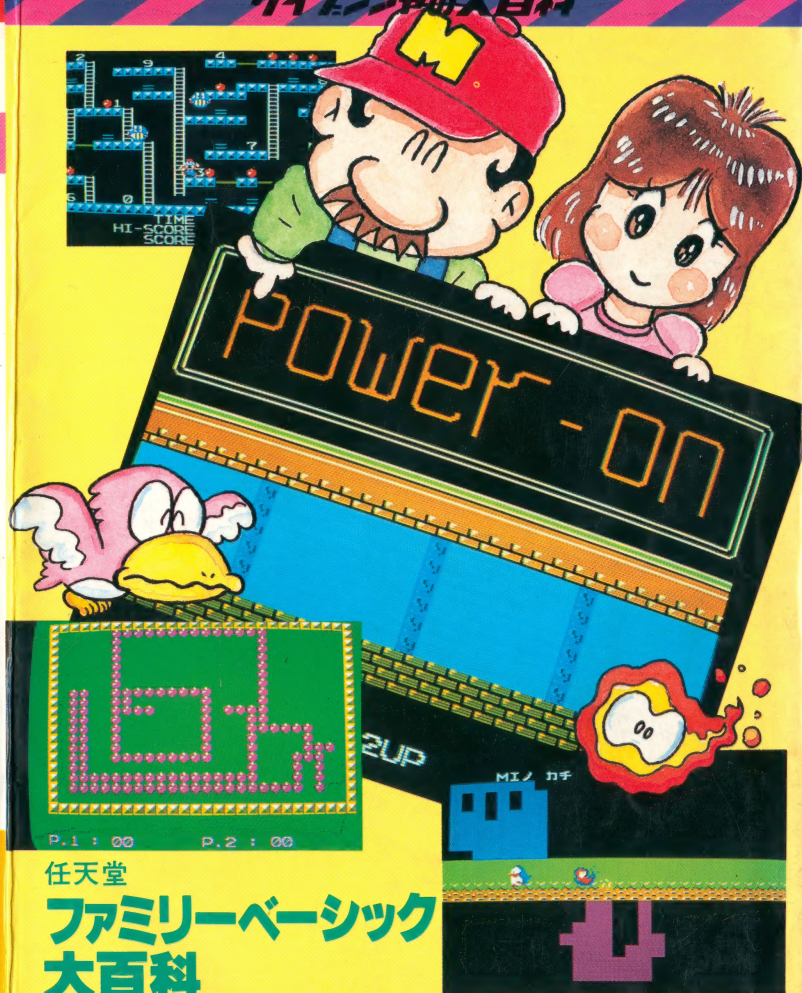
3 最新版世界の飛行機大百科 650円
8 最新改訂版野生動物大百科 650円
9 最新版自動車もの知り名鑑
大百科 650円
10 特急・私鉄大百科 600円
13 鉄道もの知り情報大百科 650円
14 最新版戦闘機大百科 650円
15 最新版特急・急行大百科 650円
17 カメラ入門教室大百科 600円
19 手作りおもちゃ大百科 600円
20 世界の鉄道大百科 600円
21 なぞなぞクイズ大百科 600円
22 動物もの知り大百科 500円
23 テレビヒーロー大百科 650円
28 宇宙大百科 650円
29 仮面ライダー大百科 600円
31 宇宙戦艦ヤマト大百科 650円
32 さらば宇宙戦艦ヤマト大百科 650円
33 パズルクイズ大百科 600円
43 カラー版昆虫大百科 650円
45 おりがみ大百科 600円
47 恐竜大百科 600円
51 推理クイズ大百科 600円
55 ヒーロークイズ大百科 600円
58 ヒーローマシーン
必殺技大百科 650円
71 拳銃・マシンガン大百科 650円
73 全私鉄大百科 650円
75 ヒーローなぞなぞクイズ
大百科 650円
78 機動戦士ガンダム大百科 650円
80 川づり大百科 650円
81 映画版機動戦士ガンダム
大百科 650円
84 カブト・クワガタ大百科 650円
85 ヒーローロボット大百科
PART 2 650円
88 手作りおもちゃパート2
大百科 650円
90 鉄道写真大百科 650円
92 映画版機動戦士ガンダムII
大百科 650円
98 宇宙大百科PART 2 650円
99 天体・星座大百科 650円
102 手品・ゲーム大百科 650円
109 電動工作大百科 650円
111 海づり大百科 650円
114 映画版PART 3
機動戦士ガンダム大百科 650円
117 プロレス大百科PART 2 650円

小さくて、あつくて、楽しい本！

ケイブンシャ 新シリーズ

ファミリーコンピュータ ★ゲーム必勝法シリーズ

1.ゼビウス
2.エキサイトバイク/アイスクライマー
3.バンゲリングベイ
4.チャンピオンシップ/ロードランナー
5.サッカー/テニス/ベースボール/ゴルフ

定価各380円　（予定）

株式会社　勁文社　発行　昭和60年6月5日　初版　定価650円

ファミリーベーシック大百科

222

任天堂 ファミリーベーシック™大百科

ケイブンシャ

ケイブンシャの大百科

任天堂

ファミリーベーシック 大百科

© 1985 Keibunsha Printed in Japan　雑誌コード63544-73　定価650円

PERFECT
SCORE
222
任天堂
ファミリーベーシック大百科
ケイブンシャ

ケイブンシャの大百科

任天堂

ファミリーベーシック™ 大百科

PERFECT

SCORE: 449

任天堂
ファミリーベーシック大百科
ケイブンシャ

定価650円